Grace Hopper and the Marvelous Machine: Lessons for Modern Technical Communicators from the Mark I ASCC Manual

2017

Jessica Meyr
University of Central Florida

GRACE HOPPER AND THE MARVELOUS MACHINE: LESSONS FOR MODERN TECHNICAL COMMUNICATORS FROM THE MARK I ASCC MANUAL

by

JESSICA J. MEYR
B.A. Honors in the Major University of Central Florida, 2010
M.A. University of Central Florida, 2010

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of the Arts in English – Technical Communication
in the Department of English
in the College of Arts and Humanities
at the University of Central Florida
Orlando, Florida

Spring Term
2017

Major Professor: Dan Jones

ABSTRACT

Women's technical writing achievements often go unrecognized, both due to the invisibility of technical writing professionals in general, and a lack of famous technical communication role models in particular. The purpose of this thesis is to analyze and present an early major work in the technical writing of Rear Admiral "Amazing" Grace Hopper, inventor of the compiler and an important figure in computer science history. Although Hopper is arguably best known for popularizing the idea of the "computer bug," her achievements in computer science extend from invention of the software compiler to tireless promotion of the programming language COBOL.

Her work *A Manual of Operation for the Automatic Sequence Controlled Calculator,* written for the first digital computer in America, is analyzed here according to Mike Markel's eight criteria of excellent technical writing: honesty, clarity, accuracy, comprehensiveness, accessibility, conciseness, professional appearance, and correctness. I also cover other specific strengths of Grace's approach, including how she establishes sufficient context, highlights multiple uses for information, and provides numerous well-chosen examples for audience needs. However, I also discuss how modern research principles for improving technical writing, including task-orientation, attention to cognitive load, and minimalism, help explain the manual's shortcomings.

I conclude my study with a discussion of Hopper's later work, "The Education of a Computer," to demonstrate her growth as a writer. The conclusion also highlights areas awaiting further research and cements my recommendation that study of Grace Hopper's work be incorporated into our historical understanding of the discipline. Hopper's technical writing deserves to be more widely understood and appreciated as a vital contribution to early software documentation.

For Matthew

ACKNOWLEDGMENTS

For their inspiration, tireless support, and years-long dedication to the cause, my thanks and all my love to my parents.

Many thanks to Dr. Dan Jones, for untiring patience and unflagging faith, resolve, and teamwork in seeing this one through.

My thanks and gratitude to Dr. JD Applen for his quick wit and ready support.

Many thanks to Dr. Paul Dombrowski for his support and insights in the work and the rework.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ACRONYMS

ASCC – more formally, “the Harvard Mark I ASCC,” the Harvard Mark I Automatic Sequence Controlled Calculator, the first American digital computer.

COBOL – Common Business Oriented Language

ENIAC – Electronic Numerical Integrator and Computer

FORTRAN – Formula Translation

UNIVAC – Universal Automatic Computer

CHAPTER ONE: INTRODUCTION

This thesis focuses on the technical writing of Rear Admiral Grace Brewster Murray Hopper, primarily *A Manual of Operation for the Harvard Mark I Automatic Sequence Controlled Calculator*, some of the first ever user documentation for an American computer (Beyer 16). This thesis also discusses a suite of three articles on the operation of the Mark I, co-written with her mentor, Howard Aiken, for *Electrical Engineering*. Hopper's later article "The Education of a Computer," is also examined, noteworthy both as a minor classic in computer science and the first documentation ever written for software compilers (for example, Beyer 16, 21, 30). There are four reasons for this focus: first, while Hopper has been the subject of considerable prior scholarship, and is a well-known figure in computer science circles—witness the popular Grace Hopper Celebration of Women in Computing conference—there is little study of her technical writing as a corpus (Williams 2). Second, to the best of my knowledge, there is no scholarship discussing Hopper's relevance as a technical communicator. In addition, the manual for the Mark I ASCC was one of Hopper's first large-scale professional documentation projects, written and edited early in her computing career: this manual is the first of her major technical documentation achievements (Williams 49). Finally, the manual itself is an important artifact in computing and user documentation history.

Grace Hopper and the Marvelous Machine: Lessons for Modern Technical Communicators from the Harvard Mark I ASCC Manual seeks to bring lessons learned from the dawn of computing to today's technical communicators, with a famed and notable expert at the helm.

Purpose

The purpose of this thesis is to achieve three objectives: introduce technical communicators to Grace Hopper's accomplishments, emphasize Hopper's work with the Mark I ASCC and confirm its place in computing history, and provide modern technical communicators with practical insights and applications for their own writing based on Hopper's manual for the Mark I ASCC and selected other writings. For example, her *Electrical Engineering* articles on the Mark I ASCC, co-authored with Aiken, provide additional user instructions and explanations for working the computer, while being geared more toward computer operators—users—than programmers. This shift in focus makes the articles distinct yet relevant objects of further study.

This thesis seeks to answer the following research questions:

- What makes the Mark I ASCC a worthy object of study? What is the least we need to know about the device to understand its manual, and vice versa?
- What makes the *manual* for the Mark I ASCC a worthy object of study as a technical document?
- What can Grace Hopper's approach to this manual, considered alongside a selection of her other writings, teach modern technical communicators about technical documentation?

Scope

This thesis is limited to an investigation of Hopper's manual for the Mark I ASCC and selected other writings: her *Electrical Engineering* articles for the Mark I ASCC, and her article "The Education of a Computer," for UNIVAC architecture. This tight focus is due in part to the fact that the Mark I ASCC manual, at 561 pages, is both Hopper's first major technical documentation effort, and a complete, substantive example of her early work. The *Electrical Engineering* articles, meanwhile, enhance readers' understanding of her skills, approach, and style, while still explicating operation of the Mark I ASCC. "The Education of a Computer," detailing similar instructions for a more advanced machine, reflects upgrades not just to computer technology but also to Hopper's approach to it. While this sample of her work is not exhaustive, it is intended as representative, a highlight reel of Hopper's technical documentation. This thesis does not examine her work with COBOL, or her lectures for the Navy in detail, as neither of these applied to her work on the Mark I ASCC. Frankly, an examination of all of Hopper's writing—both her mathematics work and numerous articles in the computing field—is a task more suited to a formal dissertation, and one that must await future research.

Methodology and Organization

I conducted the research for this thesis via a thorough investigation of interdisciplinary sources, including academic papers in fields ranging from military history to computer science (Campbell-Kelley et al.; Ceruzzi *Prehistory*; Cohen). Many of these sources were obtained through exploration of scholarly online databases, including *IEEE Xplore*, *ACM Digital Library*, and *Academic Search Premiere*. Sources such as *Electrical Engineering* in the ACM deep backfile proved of particular value in contextualizing Hopper's writing (Aiken and Hopper *ASCC I*). Other resources, such as Ceruzzi, Beyer, and Cohen, put the Mark I ASCC in its proper place societally and technologically, integrating computer science knowledge and history with historical and biographical data about Grace Hopper. These sources were used to blend qualitative data, such as colorful anecdotes taken from interview records, with hard quantitative data such as hardware specs, to illustrate the dynamic complexity of early computing and highlight insights for modern technical communicators in Hopper’s work.

I also consulted print research, including published biographies, peer-reviewed scholarly articles, prior dissertations, and articles written directly by Hopper. Moreover, I included a few modern popular articles, such as “Grace Hopper and UNIVAC: Before There Was COBOL” in *LinuxVoice*, showcasing the popular picture of Grace Hopper (Kemp). It is this writer’s recommendation that our discipline integrate Hopper as one of its previously unsung heroes. This thesis exists, in part, as a beginning effort toward doing so.

As a final methodological note, this thesis draws extensively from secondary sources for biographical data. There are a variety of reasons for this. Unfortunately, Hopper herself has been deceased for some time and thus certain direct research methods, such as e-mail correspondence or personal interviews, are impossible. In addition, little of Hopper’s personal correspondence survives, limiting study of her life in her own words (Grudin and Williams 17). Finally, her manual for the Mark I ASCC is a primary source, consulted directly. Because of my specific focus on her written work, several quality secondary sources are sufficient for establishing Hopper’s biographical data.

The organization of this thesis is fairly straightforward. This first chapter outlines the stakes and advantages of the research, and provides an overall literature review. The second chapter showcases Hopper’s achievements, such as popularizing the concept of the computer ‘bug’ and her influential work with COBOL (Beyer *Information Age*, 7; Williams 76, 77). Chapter 3 provides detailed explanation of the Mark I ASCC and its components, because they are an extinct technology; punch cards, for example, are no longer used in computing. Chapter 3 also outlines Hopper’s manual in more detail to prepare the reader for Chapter 4, which presents the strengths of Grace's writing with an eye to lessons and insights for current and future technical communicators. "The Education of a Computer", in particular, showcases the growth of her instructional writing and provides additional lessons learned. It is also the first piece of documentation ever written concerning software compilers (Mitchell 2). The final chapter synthesizes findings, attempts to address any faults in argument, and presents potential areas for future research.

Significance

This is not a biography, but instead an examination of Grace's written work. Second, to the best of my knowledge, no research to date considers Hopper's technical documentation foremost: this thesis thus addresses a critical gap in existing research into Grace Hopper's life and career as a major contributor in computing history, and it does so in terms that matter to technical communicators. Finally, the history of technical communication—while a thriving area of the discipline—does not provide enough attention to role models or specific technical writers in general, and to known female technical writers in particular. This thesis seeks to redress both issues, and, in part, establish Rear Admiral Grace Murray Brewster Hopper as a writing role model for modern technical communicators.

Literature Review

In order to understand the relevance of Grace Hopper's writing for current technical communicators, it is first necessary to examine and contextualize her work. This literature review encapsulates relevant research into some of Grace Hopper's writing milestones, and addresses the research gap in technical communication around the acquisition and canonization of good technical communication role models, highlighting the need for more female technical communication role models, in particular.

From a historical perspective, women's contributions to the field of technical communication are not widely recognized. This is due in part to the origin of the discipline: engineering English catered to the requirements of a male-dominated field (Johnson-Eilola and Selber 37; Malone *Fifteen Years*). Another reason women's achievements in technical writing may have gone unremarked is cultural: there is a general unspoken assumption that work must occur in the workplace, the traditional purview of men (Johnson-Eilola and Selber 36). Moreover, when the Great Depression placed a stranglehold on national employment, women working in mathematics and other scientific fields were encouraged to vacate their positions, freeing more opportunities for men (Williams 15). This trend increased through the following decade. As technology proliferated in complexity and scope after World War II, many working women—particularly those in scientific, industrial, or technical fields—reverted to domesticity to make room in the workplace for returning GIs (Williams 16). Grace Hopper was a notable exception, remaining a Vassar mathematics instructor during the Great Depression, and serving as a Naval officer during World War II (Beyer *Information Age* 25, 27, 35; Williams 18). Over the years, she returned to Naval service three separate times before retiring in 1986 at the rank of Rear Admiral by special appointment (Williams *xi*). In the intervening years, Hopper served as a programmer and consultant at several prominent computing corporations, and as a noted public speaker for the Navy (Beyer *Information Age* 247; Williams 91, 97).

Current historical and biographical scholarship remembers Hopper primarily as a teacher, a programmer, and a Naval officer, overlooking her important contributions to technical communication (Mitchell; Williams; Beyer *Information Age*). This thesis thus focuses on

Grace's most materially significant piece of user documentation, framing Hopper the legend in terms of Hopper the technical writer. The scope of this thesis is limited to an investigation of Hopper's work on the Mark I ASCC, particularly the manual for the device itself, and investigation of selected additional writings by Hopper to further clarify and explain her writing style and no-nonsense approach to technical information.

Women in Technical Communication

While the role of women in technical communication history remains under-examined, and there are only a few studies dedicated to the "lone heroes" of technical communication in general, in recent years there has been extensive historical research (Todd 67; Malone *Fifteen Years*). Tebeaux, in particular, has conducted studies concentrating on the history of particular document types throughout the Renaissance: she examines shipwrights' manuals, domestic handbooks, and gardening practices almanacs or manuals, explaining that these "domestic how-to books" are some of the earliest technical writing by and for women (Tebeaux and Moran 211; Tebeaux and Lay 196). Shirk has also examined the contribution of female technical writers to 18th century botany (1997). Durack's history of early sewing machine manuals also delivers an investigation of the rhetoric, persuasive appeals, and sexist language and imagery used to market the device (*Sewing Machine* 1998).

The domestic focus of many of these fields and technologies is partially an artifact of traditional gender roles: according to Durack, "history in general, and the history of technology in particular, have tended to omit the activities of women in part by locating significance" primarily in public and political arenas, traditionally the purview of men (in Johnson-Eilola and Selber, 36). Because of this rhetorical climate, "their work was going virtually unnoticed because of the lack of written accreditation based solely on gender," and studies on female technical writers who are also “lone heroes” of technical communication are almost nonexistent (Rauch 400; Malone *Fifteen Years*).

In fact, other than this thesis, the only such study of which I am aware is Rauch’s *The Accreditation of Hildegard von Bingen as Medieval Technical Writer*, examining the medical texts and health communication of Hildegard von Bingen—traditionally conceptualized as a poet, visionary abbess, and “musicologist,” she also wrote extensively on everyday pathology and kept patient records strikingly similar to modern physician’s reports (397, 403). While intriguing, this is just one study; while significant work has been done establishing the historical relevance and impact of women technical communicators, much more remains to be done.

Manuals are a frequent and favorite topic in historical technical communication research (Tebeaux and Moran 62, 65). Technical manuals in English typically trace their lineage from Chaucer’s *Treatise on Astrolabe*, and the lucid, direct “plain language” technical writers seek to cultivate is often attributed to English seamstresses’ manuals (Rauch 400; Tebeaux and Moran). Ramey’s *The Coffee Planter of Saint Domingo: a Technical Manual for the Caribbean Slave Owner* (2012) brings critical theory to the study of technical manuals, illustrating that a technical document may be factually complete, precise, detailed, and well-written, yet still serve odious rhetorical or social purposes—in this case, legitimizing the French colonial slave trade of the 18th century (144). No technical document is completely neutral; while hardly in the same vein, Hopper’s manual also functioned as PR for the Mark I ASCC and the Harvard Computation Lab (Beyer *Information Age* 137).

In terms of somewhat more modern American technical communication, much has been made of Benjamin Franklin’s technical works, including his extensive diagrams for the Franklin stove, and the 1912 transliteration of *De Re Metallica* by Herbert Hoover (Todd 70, 71; Voss

241). Focusing on Grace Hopper's manual for the Mark I ASCC further establishes an American tradition of technical communication and serves at least two additional purposes: Todd suggests that "name recognition" bolsters the historical context of technical manuals, and "provides students with historical examples of the writing they will do after graduation" (70). Given that specific technical writing achievements of particular women remain underrepresented, and that role models are an important, if also rarely examined, factor in historical technical communication research, Grace Hopper's *Manual of Operation for the Harvard Mark I Automatic Sequence Controlled Calculator* is a worthy object of study, just as Hopper herself is a worthy subject.

Technical Communication: What's In a Manual?

According to Rubens, Hopper's documentation for the Mark I ASCC is best categorized as a user reference manual, "a compromise between a user guide and a reference manual" (7). The chief distinction here is purpose-based: user guides focus on helping people complete specific tasks, while reference manuals describe the entire context of a product, process, or service (Rubens 5, 7). User reference manuals are designed to impart both short- and long-term knowledge, and "describe […] everything users can do with a system," often because time or budget constraints prevent the completion of separate, shorter manuals tailored to individual audiences (Rubens 7, 16). The Mark I ASCC was built during wartime for use by scientists and mathematicians, and considered a national defense asset: Hopper's intended audience was, necessarily, a specialized one (Rubens 7, 16; Beyer 124). At the time, computers were a technological breakthrough that required a great deal of conceptual information and background (Rubens 5, 13, 17). While the general knowledge level of Grace's audience could be safely estimated as quite high, these “subject experts” also required clear explanation of novel processes and procedures, such as coding a sequence tape, or knowing how frequently computer operations should be checked: Hopper recommended intervals of no more than twenty minutes’ continuous operation (Rubens 7; Hopper et al. *Mark I Manual* 111).

This driving audience need for additional context meant Grace included quite a bit of reference information before any operational section of the manual (for example, Hopper et al. Hopper et al. *Mark I Manual* 80-95). Because of this need for context, the entire first chapter of the manual is devoted not to machine operations or discussion of components, but to an absorbing history of the mathematical devices that led to development of the Mark I ASCC, from the abacus to Babbage’s difference engine (Hopper et al. Hopper et al. *Mark I Manual* 1, 4). Such reference material is of interest to the intended core audience, even where it does not help them solve a specific problem, while operators interested in resetting the plugboard, replacing a relay, or debugging their code could safely proceed directly to procedural data in later chapters (Rubens 16).

The hybrid purposes of a user reference guide are reflected throughout the document. Grace’s first explanation of any problem is quite in-depth, often approaching narrative style, and is probably intended to be referred to only once—standard characteristics of reference information (Rubens 5, 6, 16). Further problem examples, designed to be referenced repeatedly, are brief and operational in nature (Rubens 6, Hopper et al. Hopper et al. *Mark I Manual* 197, 201). Larger procedural sections intended for reuse, such as Chapter IV, Coding, contain reference features that aid users seeking a specific solution (Hopper et al. Hopper et al. *Mark I Manual* 98; Rubens 13). For instance, a mathematician who wanted the Mark I ASCC to generate logarithm tables would refer to page 98, based on the main table of contents, and then to page 162 from the inset table provided (Hopper et al. *Mark I Manual* 98).

More than just a clearly designated user reference guide, Hopper's manual also embodies particular criteria of excellent technical writing (Markel 2012). Based upon Markel, there are eight dimensions of excellent technical writing: honesty, clarity, accuracy, comprehensiveness, accessibility, conciseness, professional appearance, and correctness (14). Honesty is being both truthful and ethical with the audience: "deliberately omitting important information can defraud, injure, or kill" users (13). For Markel, clarity is avoiding redundancy, while accuracy is an absence of agenda or "spin" in the document (14). Comprehensiveness "provides all the information readers need" in terms of examples, data, and concept coverage: enough for a novice, without weighing experts down (14). Accessibility is Markel's term for ease of use and clear document organization; "readers should not be forced to flip through [unnecessary] pages" (14). Conciseness is use of plain language, while professional appearance and correctness refer to visual and stylistic integrity: "if the document looks neat and professional, readers will form a positive impression of it," its author, and its content (13, 14).

Grace Hopper's manual distills at least four of Markel's principles—clarity, accuracy, comprehensiveness, and correctness—and so qualifies as an example of excellent technical writing. Certain shortcomings of this manual, meanwhile, are valuable to technical communicators seeking to hone their skills. For example, accessibility in the manual could have been improved with the use of topic subheadings (Rubens 32; Markel 50). These and other insights into the manual itself will be covered in more detail in Chapters 3 and 4.

Grace Hopper: Computing Pioneer, Technical Communication Powerhouse

When she entered the computing field in 1944, Hopper was already an accomplished mathematician and Vassar instructor (Williams 13, 18). Today, Hopper is arguably most famous for her role in a maintenance incident with the Mark II in 1947: when the computer crashed late one night, the Mark II team discovered that a large moth had jammed itself in the relays (Beyer *Information Age* 65). Hopper's team removed the offending insect and taped it to a card with the inscription "first actual case of bug being found"; Grace is thus usually credited with "debugging" the machine (Beyer *Information Age* 65, Williams 38). "Bugs" were, at the time, engineering slang for any mysterious mechanical errors, but Hopper popularized "bugs" and "debugging" for computer code specifically (Kemp 83). She is also the inventor of the A-O compiler (Kemp 83). However, Hopper's specific programming and documentation work are less well-known outside of computer science circles (Grudin and Williams 17, 18). I seek to address this knowledge gap by focusing on an extensive example of Hopper's technical documentation: the manual for the Mark I ASCC.

The inventor of the first working software compiler, Hopper was also a key developer of COBOL, and the original programmer of related ancestor languages such as FLOW-MATIC, originally written to compete with FORTRAN (Kemp 84; Williams 87; Beyer *Information* Age 277, 289). While "recognized in her lifetime with 'most of the honors that can be given anyone in the computer industry,'" Hopper herself remains somewhat inscrutable, perhaps due in part to a lack of surviving personal correspondence or papers, and a persistent gender disparity in computer science (Hopper, "Future Directions"; Grudin and Williams 17). If researchers are to form a more complete picture of Hopper as technical communicator, her writing must be studied directly, which is the chief aim of this thesis.

Technological Dead-End, Conceptual Marvel: The Mark I ASCC

Because punched card computers are obsolete, it is necessary to explain the Mark I ASCC to readers before talking about Hopper's manual for it in depth. Radically different from what modern audiences would recognize as a computer, the Mark I ASCC resembles a huge abacus that displays numerical values by physical movement of individual counters from 0 through 9, much like an odometer (Beyer *Information Age* 47; Ceruzzi *Prehistory* 149, 150).

The principles behind the Mark I ASCC were originally developed by Howard Aiken in 1937 (Beyer *Information Age* 36). While working on his doctoral thesis in physics, Aiken realized certain calculations were simply beyond him: the sheer volume of arithmetic needed to solve differential equations and series particular to his work required more math than a human being could complete in years, decades, or even a lifetime (Beyer *Information Age* 37). Aiken needed mechanical assistance to offload the sheer volume of data involved, and knew that it would take something more accurate, and a great deal more powerful, than a desktop adding machine (Beyer *Information Age* 37; Cohen 34, 36).

The most impressive American computer of its time, and the first American digital computer, the Mark I ASCC filled a room end-to-end and completed three calculations per second (Beyer *Information Age* 48). It contained 530 miles of cable and 3,500 relays, and completed millions of calculations in its long career, from its first use in 1944 up through the late 1950's (Beyer *Information Age* 48, 49; Ceruzzi *Prehistory* 160). Interestingly enough, the Mark I ASCC was already a technological dead-end by the conclusion of World War II: in terms of physical construction, the Mark I ASCC had been supplanted by radically different designs in superior materials on a much smaller scale (Beyer *Information Age* 120). ENIAC and the British

UNIVAC used vacuum-tube technology for memory storage, instead of Aiken's electromechanical relays (Beyer *Information Age* 158, 159). Thus, the Mark I ASCC was considered something of a brute-force dinosaur within just a few years of its construction (Beyer *Information Age* 159, 160). However, in terms of programming, modern computers owe a lot to the Mark I ASCC: it was the first computer capable of processing commands in sequence without additional plug-and-play from an operator (Beyer *Information Age* 119, 120; Ceruzzi *Prehistory* 157).

In terms of architecture, the Mark I ASCC introduced the important concept of "end-around carry," a process by which calculations were completed by adding and subtracting on the complements of given values (Hopper et al. *Mark I Manual* 4). This principle is still used in binary sums, the backbone of all modern machine languages (Ceruzzi *Prehistory*). Thus, while obsolete almost immediately after it was built, and antiquated in both design and mechanics, the Mark I ASCC still has much to teach modern computer users and the technical communicators who write for them, especially those not proficient in computer science. A more detailed explanation of the specs, capabilities, inner workings, and hardware of the Mark I ASCC is provided in Chapter 3 of this thesis, in order that certain instructions referred to in Chapter 4 will make sense.

The Mark I ASCC Manual: Historical Lessons for Modern Technical Communicators

Of primary concern are potential lessons, insights, and tips that modern technical communicators might glean from this manual. Hopper's presentation of the material, her writing style, and her overall mindset as a technical communicator will also be examined.

One thing Hopper does exceptionally well is to provide adequate context: faced with the daunting task of describing a completely new technology to a specialized audience, she first traces the historical roots of Mark I by explaining its relationship to features familiar to mathematicians of the day, including Pascal's adding machines, Napier's bones, Babbage's non-working models for difference and analytical engines, and the Mark I's resemblance – at least structurally – to an enormous abacus (Hopper et al. *Mark I Manual* 3, 4).

While current technical documentation wisdom teaches that overstuffed history sections confuse new users and are likely to be avoided by readers, Hopper was writing about a machine that had never before existed in America (Beyer *Information Age* 129). The chief take-away for technical communicators here is not to include a large history in every manual or help file, but rather to be sure that the audience is provided adequate context for the device, using terms, examples, and concepts with which they are already familiar.

Another strength of Hopper's writing is in how she looks ahead to serve users' future needs. For example, when discussing how to program the Mark I for multiplication, she refers users to Chapter IV on "interposed operations" as a means of reducing processing time, pointing out helpful data without junking up the instructions at hand (Hopper et al. *Mark I Manual* 23). Further possibilities for interposing are later discussed in Chapter V, Plugging Instructions (Hopper et al. *Mark I Manual* 245). This is reflected in the modern technical communication dictum to try and anticipate the user's needs when writing a manual or help file.

A third strength of Hopper's manual is implied by style, and may be invisible to current technical communicators: she uses genderless language, referring throughout to "the operator" or "personnel" (Hopper et al. *Mark I Manual* 245, 289). Hopper renders most instructions in passive voice—"quantities standing in the switches must be printed or punched for checking, either under control of a sequence tape or under manual control of the keyboard"—to remove the notion of gender entirely (Hopper et al. *Mark I Manual* 289; Einsohn 413). While genderless language—a form of bias-free writing that results in such constructs as "the operator" and the dreaded "he or she"—has come under fire in recent years, there was only one alternative for it in the 1940's: "he," "mankind," "man's achievements" (Einsohn 413, 414). Hopper's consistent use of bias-free language produced a more neutral document. While she by no means welcomed "women's lib" with open arms, nor did she consider herself a feminist, Hopper at least wrote in terms that did not deliberately exclude women (Beyer *Information Age* 212; Williams 75). That technical communicators are now able to speak of including women, rather than merely not excluding them, is a mark of how far we have come—and of how far ahead of the curve Hopper herself may have been in terms of dealing with a technologically-forward audience.

Furthermore, Hopper's *Electrical Engineering* articles diversified instructional material for the Mark I ASCC: co-authored with her mentor Aiken, these articles briefly explained key concepts from the manual, with specific instructions for computer operators, separate from the mathematicians writing code for it (Aiken and Hopper *ASCC 1* 388). Written for electrical engineers—read: users instead of programmers—these articles appeared after the manual was published, and exposed the Mark I ASCC to a wider audience (Aiken and Hopper *ASCC 2* 453). Hopper returned to generalized explanations of specialized devices in her later article "The Education of a Computer", explaining UNIVAC architecture. Regarded as a minor classic in its own right, "The Education of a Computer" will be analyzed with the *Electrical Engineering* articles to yield a fuller picture of Hopper's overall writing style and approach in tackling complex technical concepts.

In order to provide a strong context for the importance of Hopper's work as a technical writer, a pedigree of her achievements is presented in the next chapter. This section will illustrate ways in which Grace's background and expertise as a mathematician, a teacher, and a Naval officer helped prepare and inform her approaches to complex technical topics and a to her instructions for a bold new American invention: the digital computer.

CHAPTER TWO: AMAZING GRACE

"Grace was a good man!"

- Howard Aiken, Inventor, Harvard Mark I ASCC

The 1940s Navy only admitted women to deal with a drastic personnel shortage, and civilian prospects for women after the war were hardly rosy: the IBM corporate culture of the 1950's was "an elite male fraternity" (Williams 12; Beyer *Information Age* 5). In this competitive environment, Hopper distinguished herself as a dedicated, chain-smoking, take-no-prisoners programmer and tireless promoter of improved software coding standards (Beyer *Information Age* 52, Mitchell 11). Many people know of "Amazing Grace" via a familiar bit of computer science folklore: popular legend has it that she invented the phrase "to debug" a computer, one muggy night in 1947 when her team discovered that a moth had flown into the relays of the IBM Mark II, causing a literal computer crash (Beyer *Information Age* 1, 64). In point of fact, though Hopper was responsible for removing and documenting the moth, the idea of "debugging" a computer was already in use via engineering parlance of the day (Beyer *Information Age* 64, 65; Williams 77). Hopper's achievements in the field extend far beyond a catchphrase. This chapter demonstrates a select few of Hopper's achievements in order to establish her as a noteworthy figure in her field, and to further lay the groundwork for discussion of her technical writing achievements in context. This chapter also demonstrates how Grace Hopper developed the specialized knowledge that enabled her master invention—the software compiler—which in turn is the target subject of her some of her most widely read later work (Beyer 322). In short, this chapter answers the inquiry: what makes Grace Hopper a worthy subject of technical writing study?

Later one of the first programmers in America and a computing legend in her own right, Hopper first distinguished herself as a Vassar mathematics professor (Williams 13). She was the only woman awarded a doctorate in mathematics during the latter half of the 1930's, and one of a handful of female mathematics professors at a time when women were vacating the workforce to increase job opportunities for men during and after the Great Depression (Williams 14, 15). While not herself a feminist, Hopper encouraged other women to continue their studies and adopted a variety of nontraditional techniques to keep students eager and engaged with the material (Williams 15, Mitchell 19). Hopper revolutionized the mechanical drawing curriculum, a piece of coursework she described as "deadly," by borrowing concepts and principles from animation—Disney had only recently debuted *Snow White and the Seven Dwarfs*, and the topic was a hot one (Williams 16, 18, Beyer *Information Age* 27). Hopper packed classrooms that before had been nearly empty and reached students from a variety of disciplines. In engineering math courses, she made ballistics problems intriguing by turning the usual bullet examples into problems involving rockets—devices she would later solve firing tables for on America's first digital computer (Williams 50, Beyer *Information Age* 29). During her time at Vassar, Hopper honed her enthusiasm for teaching, and correcting students' proofs improved her own instructions, which would later prove valuable during her time working with the Mark I ASCC (Beyer *Information Age* 32).

Despite her civilian success as a mathematician and instructor, Hopper's Naval career had a rocky start: due to a scheduling error, her commanding officer Howard Aiken had expected her the week prior, and when she reported for duty, he demanded to know "where the hell she had been" (Williams 24, Mitchell 27). Aiken handed Hopper a stack of punch cards and a slim book of program codes and informed her that she could find a place to live tomorrow; right now, he “would be delighted to have the interpolation coefficients for the arc tangent series,” and he expected it by the end of next week (Williams 26). Hopper, rattled but undaunted, learned to code for “the beast” of a machine in record time, and came to thrive under Aiken’s take-no-prisoners tutelage (Williams 26).

Hopper worked most closely with fellow coder Robert Bloch, who was more of a theorist than an engineer— she said “it drove us all crazy that he could” write a program correctly the first time, longhand, where Hopper herself frequently built from subroutines she had already written and proven (Williams *Information Age* 48). Together, they surmised correctly that one particular problem set had to do with “atomic fission” in 1943: known to them only as Problem K, it was later declassified as part of an implosion envelope study for planned atomic yield at Hiroshima (Williams 46). Hopper’s direct, solo programming efforts for the Mark I ASCC included proximity values for deployment of underwater mines, as well as firing tables for self-guided rockets, which were then a brand new invention (Williams 48, 49). She would not begin writing the Mark I ASCC user manual until well into her first year on the project, though of course she and the other coders kept notebooks of useful subroutines and commands in order to save all-important computing time (Beyer *Information Age* 45, 52).

Hopper is arguably most famous as the inventor of the software compiler (Kemp 83). In many ways, Hopper’s work on the Mark I ASCC provided the necessity that was the mother of her invention: she had particular difficulty writing software loops, or steps that told the computer to perform certain actions more than once (Williams 52). Suppose a sum needed to be added a dozen times. Often, Hopper could not recall if she had begun a program on step one or step zero, so the computer might repeat the sum eleven or thirteen times (Williams 52, 53). She was not alone in this particular bugbear: her coding partners, Bloch and Campbell, also fell victim to the tedium of writing lengthy, repetitive series of commands by hand (Beyer *Information Age* 98). Because data and parameters had to be fed manually to the tape punch, the potential for human error loomed in all calculations and worsened further down the program: the computer would automatically repeat its erroneous instructions until cued to stop (98). While codebooks did keep the team from constantly reinventing the wheel, Hopper was certain the computer could be made to automate more of its coding work, though the exact method would elude her until her later work for Remington Rand (Beyer *Information Age* 221). She was chasing the vision of "intuitive, user-friendly, hardware-independent" instructions that could generate efficient working programs of their own, something still not fully realized today (Beyer *Information Age* 229). In 1952, this vision yielded Hopper's A-0 compiler (Kemp 83). A compiler is a program that collects subroutines from a preset library and builds them into a program; software compilers "turn your source code into a program" (Liberty 11, 15). Without compilers, computers as we know them would not exist, and programming would be much more complicated (Liberty 11, 14, 16). While detailed coverage of compilers is beyond the scope of this thesis, there can be no doubt that Grace Hopper revolutionized modern computing via their creation.

Hopper is also credited as a pioneer of COBOL (Common Business Oriented Language), a programming language still "spoken" by 95 percent of major banking mainframes in the United States (Beyer *Information Age* 302, 303). COBOL was one of the first computer languages to use recognizable English words to represent particular commands, including "ON," "COMPARE," and "DIVIDE A INTO B," as well as the Boolean logic familiar to most of us today in conducting keyword searches, such as "OR," "AND," and "IF/THEN" (Beyer *Information Age* 303, 304; Sammet 126, 127).

Hopper's primary role in the success of COBOL was not inventing it, for that was a group effort decried by early critics as "design by committee," nor in its development—again, a team effort, one spearheaded by Hopper's protégé Betty Holberton and Jean Sammet, among others—but Hopper's driven and tireless promotion of COBOL to any interested party (Beyer *Information Age* 305, 306; Mitchell 48, 50). Hopper was the foremost marketer of COBOL, pursuing Naval and commercial audiences alike. As a result, COBOL is computing's oldest surviving language, except perhaps FORTRAN, and still one of the most recognizable (Beyer *Information Age* 280; Kemp 82). While not directly related to instruction and not technical in focus, Hopper's work with COBOL is no less important: the success of any invention "does not end with production of a prototype" (Beyer *Information Age* 306). Technical communicators are rarely granted the opportunity to appreciate the full fruits of their work in such a practical, concrete way, but Hopper achieved just that (Beyer *Information Age* 292). Ironically, Jean Sammet, one of the primary writers of COBOL, later admitted that "had [we] realized at the outset that the language we created was going to be in use for such a long period of time, we would have gone about it quite differently"; the initial release of COBOL was not "intended for longevity" (Sammet 125).

With such an array of accomplishments available for study, why focus solely on Hopper's technical writing? Several factors make Hopper's technical writing a vital part of her legacy that should be more widely studied. First, to the best of this researcher's knowledge, such a study has never before been attempted. While much has been made of Hopper's contributions to computing (witness the Grace Hopper Celebration of Women in Computing conference) and her educational and public speaking prowess, no prior scholarship covers the nature or influence of Hopper's writings directly (Alvarado and Judson 70; Mitchell 11; Beyer *Information Age*). One reason for this is that despite a thorough collection of articles, memos, and even a “humor file” kept by Hopper herself, records of her personal correspondence are almost nonexistent, making a more typical narrative history somewhat difficult (Beyer 85, Mitchell 9). Scholarly emphasis on Hopper’s technical writing precludes this issue, making it possible to evaluate her work on its own merit.

Moreover, consider that a discipline thrives on its heroes, and the heroes of technical communication remain largely unsung (Malone *Fifteen Years*). In terms of the study of technical communication history, much attention has been paid to the documents of particular eras, such as the Renaissance, and to different types of technical documents, including manuals covering a variety of topics, from strip mining to sewing machines (Moran and Tebeaux 76, 77). However, this extensive coverage often does not include any sense of authorship. Only rarely do technical communication histories address the idea of notable figures in our discipline. In particular, there is little focus on the idea of “the great writer” or lone heroes of technical communication (Malone *Fifteen Years* 334; Moran and Tebeaux 76). Where physicists have Feynman, Einstein, and Hawking, and scholars of literature can chart a legacy back well before Chaucer, technical communicators still have room to discover the champions of their discipline.

Good exploratory work has been done. Histories establishing the origins of technical communication have investigated such figures as Mark Twain, Edgar Alan Poe, and Herbert Hoover as technical communicators (Malone *Fifteen Years* 334). A rhetorical link between their status as “great writer[s]” is both implicitly and explicitly established in a variety of ways—aimed always at associating them with technical communication, conferring that same greatness upon the discipline (Malone *Fifteen Years* 334). It is just that there remains more work to do.

Therefore, when seeking out technical documentation heroes, it helps to include one who literally wrote the book on programming at the beginning of the computer era.

Grace Brewster Murray Hopper, (Rear Adm USN) is not just a computer science maven, but also one of the great unsung technical writing heroes. Without prior experience, she drafted the entire manual for a machine that had never before existed, and she did so at a rate of nearly five pages per day (Beyer 123; Mitchell 33). Hopper also oversaw all circuit diagrams for plugging and coding of the machine, and drew many of them herself, though she was not in charge of graphics or figures for the manual (Williams 50, Hopper et al. *Mark I ASCC*). She also wrote most of the code in the coding section (Beyer 60, 61). In short, it could be argued that Grace Hopper is responsible for some of the first ever software documentation in American history.

Besides her landmark manual for the Mark I ASCC, Hopper is also celebrated in computer science circles for her work with UNIVAC, particularly the seminal paper “The Education of a Computer,” which describes how to “teach” a compiler its work (Mitchell 28). It is time that this document was more widely read, or at least more widely understood, as a technical writing milestone: it is among the first series of operator instructions that resemble the software manuals we are familiar with today (Hopper “Education”). This thesis draws on material and insights from “The Education of a Computer” where applicable, because such comparison helps clarify Hopper’s mindset and approach as laid out in the manual for the Mark I ASCC, and also to contextualize the lessons future technical communicators may learn from the work of “Amazing Grace.”

CHAPTER THREE: THE MARVELOUS MACHINE

Because the Mark I ASCC is a notable relic almost unrecognizable as a computer to current audiences, it is necessary to explain the ASCC itself in more detail before discussing Hopper's manual for it. Technically obsolete as soon as it was built, the first American digital computer stood a class apart from the other massive postwar mainframes: the Mark I ASCC used the same logic as a modern computer—it could "think" like modern computers, complete certain commands on its own, and used programming concepts we are familiar with today, albeit in more primitive form (Beyer *Information Age* 119, 120). This chapter explains how the Mark I ASCC was developed: what it was made of, how its main parts functioned, and what types of problems it was designed to solve. Because the Mark I ASCC is extinct hardware, these explanations represent the least readers need to know to understand Grace Hopper's instructions for the machine, and thus how her writing may benefit current and future technical communicators.

The Mark I ASCC was the first American programmable computer: assuming all data were inputted correctly, it followed an entire tape of instructions until cued to stop (Williams 30, Beyer *Information Age* 121). IBM unveiled the Mark I ASCC to the world on August 7, 1944—by which time it had secretly been crunching numbers for the Navy for almost a year (Ceruzzi 134). Though Aiken had courted Harvard as a possible developer for his "Proposed Automatic Calculating Machine" as early as 1938, it ultimately took both his Naval connections and the advent of World War II to make his thesis-fueled dreams a reality—an adding machine that could think for itself, an IBM-funded, five-ton monster the press dubbed "Harvard's Mechanical Brain" (Mitchell 26).

While most Mark I ASCC hardware was composed of existing IBM products, such as card punches and electric typewriters, Aiken was the first person to propose, design, and build a computer using this hardware (Beyer *Information Age* 36, 121). As the Mark I's inventor, he is sometimes also credited as the sole author of its documentation, which is incorrect (Ceruzzi 146, Beyer 121). He was, however, instrumental in getting the manual written: while Hopper had published extensively at Vassar and conducted intensive workshops as a professor, she was taken aback by such a tall order, protesting that she "had never written a book!" to which Aiken laconically replied, "You're in the Navy now!" (Mitchell 33; Beyer *Information Age* 123).

Programming the Mark I ASCC was the end goal of Hopper's intended audience and one her manual served throughout. It also functioned as a form of public relations, "to educate a wider audience about computers [and] highlight the heroic achievement of Aiken and his crew deep in a basement at Harvard during the war" (Beyer *Information Age* 138). In part to further that impression, Hopper and Aiken also co-wrote three articles for *Electrical Engineering* summarizing the manual's key points while emphasizing the Mark I's features and technical capabilities (Hopper and Aiken *ASCC I*). These articles, along with a flurry of press pieces such as "Highbrow Harvard Bows to a Robot Brain," spurred civilian commercial interest in the Mark I ASCC (Beyer *Information Age* 91, 92; Williams 228). Companies such as General Electric and Eastman Kodak formally inquired about computing some of their own engineering problems, and Hopper's father—a renowned New York insurance magnate—wrote to her excited about the computer's potential for "keeping records, calculating premium tables, and generating premium and billing statements" (Beyer *Information Age* 91, 92). However, Aiken believed the chief purpose of his computer should be "to promote science" in an academic sense, and in any case, his crew was much too busy solving urgent ballistics and submarine navigational problems for the Navy. Thus, interested civilians corporate and citizen alike received form letters concluding that "perhaps, at some future date, after the war, we may be of service to you" (Beyer *Information Age* 91, 140).

Laboring in the Shadow of ENIAC

The lofty intellectual views of its creator, combined with the wartime need for utter military secrecy, may partially explain why the Mark I ASCC is usually an historical footnote beside the more famous ENIAC (Electrical Numerical Integrator and Calculator). Personal and organizational difficulties may also have factored in: Aiken was famously abrasive and feuded bitterly with IBM executive Thomas Watson throughout the Mark I project and the careers of both successor machines, the Mark II and Mark III (Beyer *Information Age* 109, Williams 35). However, the most likely reason for ENIAC's fame is its superior processing speed: ENIAC ran instructions a thousand times faster than the Mark I ASCC (Beyer *Information Age* 8; Ceruzzi 157, 266).

Despite this glaring disparity in speed, the Mark I ASCC did what ENIAC failed to do: it ran almost nonstop (Beyer *Information Age* 144; Williams 35, 37). Mark I ASCC hardware was based upon the proven technology of telephone wiring; its relays were more reliable than ENIAC's vacuum tubes and also easier to service parts for in wartime. During materiel shortages, technicians were able to fit the Mark I ASCC relays with piano wire instead of costly brass needed for munitions (Williams 36, 50; Beyer *Information Age* 68). By contrast, ENIAC's vacuum tubes had a half-life of only a few hundred watt-hours per bulb and generated so much heat that they required cumbersome water cooling (Beyer *Information Age* 69; Williams 57).

ENIAC's exceptional speed also came with a labor cost: because it processed information at near-light-speed, it could not be programmed via tape. ENIAC had to be rewired each time a different problem or solution was required (Beyer *Information Age* 51). It also had considerably less memory capacity than the Mark I ASCC (Williams 57). “Essentially,” Hopper recalled, “[with ENIAC] you built a special computer for each job, and we were used to the concept of programming and controlling it by our program” (Beyer *Information Age* 52). Thus, while ENIAC captured a place in the textbooks, the Mark I ASCC was the first American computer to both *think* and run like a modern computer. Though both machines were crucial to the war effort, the Mark I ASCC outlasted its faster, more famous cousin by four years before being decommissioned in 1959 (Ceruzzi 148).

Mark I ASCC Specs and Capabilities

The Mark I ASCC was eight feet tall, fifty-one feet long, and weighed over 8,000 pounds (Williams 29; Ceruzzi 148; Beyer *Information Age* 37). A conveyer-fed punched card machine, it understood only numbers and mathematical signs (Mitchell 26). Operated with 1,400 switches, more than 3,500 relays, and hundreds of miles of cable, it had over half a million moving parts that delivered output to either two card punches or one of two electric typewriters (Beyer *Information Age* 47, 60; Ceruzzi 148). Estimates for the final cost of creating, shipping, and assembling the computer range from $350,000 to $750,000 in 1940s currency (Beyer *Information Age* 136; Ceruzzi 148).

Programming: Problem-Solving on the Mark I ASCC

In physical terms, the Mark I ASCC was like an enormous player piano for solving equations, wired to a huge switchboard that stored its intermediate values and complex operations (Beyer *Information Age* 47; Williams 30). The computer was originally designed to solve differential equations—a problem type that helps explain how it worked in general, and in particular how it worked on other complex equations, such as the extensive tables of Bessel functions for which the Mark I ASCC earned the nickname "Ol' Bessie" (Beyer *Information Age* 113). These function tables took almost two years to complete, and were used in everything from radio signal analysis to calculating the heat resistance of various materials for weapons research and development (Williams 46). Differential equations, conceptually similar to Bessel functions, were Aiken's original source of inspiration for the Mark I ASCC, and are also a good example of how the Mark I ASCC computed solutions: by breaking large values and involved equations into small, repeated basic arithmetic that could be automated by the machine and checked by personnel.

As Ceruzzi explains, solving a differential equation first requires graphing the equation's curve and then calculating the area underneath it (*Prehistory* 136). For example, "the area under the curve given by y = sin (X) is found by evaluating the equation y = -cos (X_1), where X_1 and X_2" stand for the endpoints of the interval (136, 137). Aiken's versions of these equations were so complex they could only be solved by "breaking up the area under the curve into many small rectangular 'slivers,'" finding the individual area of each sliver, and then summing them all up. It was a tedious, labor-intensive project prone to a high degree of human error. Solved this way, the equations would have taken Aiken months or years to complete, even with the aid of a desk calculator (Ceruzzi 136, 137). By contrast, the Mark I ASCC required no more than 90 seconds to find each rectangular area and about 6 seconds per addition to tally them up (Hopper et al. *Mark I Manual* 51). While slower than today's pocket calculators, the Mark I ASCC was a significant scientific and technological leap forward (Beyer *Information Age* 11).

On its own, the Mark I ASCC could only add and subtract—albeit 23 digits at a time. Separate units for multiplication, division, logarithms, sines, and exponents all had to be connected by plugboard (Hopper et al. *Mark I Manual* 21, 246). Aiken's darlings, the interpolator units, also attached by plugboard and estimated values for functions and series, including differentials (Hopper et al. Mark I Manual 39, 246). Interestingly enough, while there was an exponent unit, there was no square root unit, a ubiquitous feature of pocket scientific calculators today (Ceruzzi *Prehistory*).

The Mark I ASCC also had no central processing unit: “Harvard’s Mechanical Brain” had no brain of its own in the modern computing sense (Ceruzzi 152). Instead, numbers inputted into the computer via tape and cards were conducted across electrical relays that read values into and out of the central bus (“buss”) and signaled the counter wheels to turn a certain number of times (Hopper et al. Mark I Manual 15, 19). In terms of the *Manual of Operation*, a relay is the wired contact transmitting data, a counter is the flywheel that displays values and counts up or down to the solution, and registers, also called “switches,” store values and commands for use in a specific relay, or for moving into and out of the central bus (Hopper et al. Mark I Manual 11, 12; 14). Though these distinctions seem obvious on the surface, the manual refers to switches and registers interchangeably with accumulators—and accumulators work quite differently in a modern machine (Ceruzzi *Prehistory* 152). Without delving too deeply into details, these basic hardware distinctions are necessary for understanding certain Mark I ASCC instructions. It helps to think of relays as wiring, and counters as both storage and display; switches and registers, meanwhile, almost always serve storage functions in Mark I ASCC instructions and programs.

The manual also details several Mark I ASCC hardware improvements developed by the crew. For example, counters 64 and 65, as well as counters 68 and 69, could be “ganged” together to double the computer’s digit capacity from 23 places to 46. This was useful in matters of scientific notation and high-accuracy computations, which dealt with very large and very small values, respectively (Hopper et al. Mark I Manual 20). Meanwhile, register 71 “could be split in half and used as two 12-digit registers" for statistics problems and in other situations requiring very big data sets that were "low[er] in accuracy" (Hopper et al. Mark I Manual 20, 21).

Programming: How It Worked

The first programmers in American history were, in order, Richard Bloch, Robert Campbell, and Grace Hopper. They worked first on paper, breaking equations into basic arithmetic steps before transposing them to tape and cards (Beyer *Information Age* 7). Hopper liked to say that "you simply step by step told the computer exactly what to do," though this understatement often represented hours or days of preparatory work (Beyer *Information Age* 53, Williams 30). To maximize efficiency, the three programmers kept a "codebook" of proven subroutines and techniques for completing a job in less time (Beyer *Information Age* 61). Programs that worked could then be copied or inserted into the new routine.

Interposition and rounding results to a desired level were especially important on Problem K, the solution of the implosion envelope for the atomic bomb deployed over Hiroshima: "I guess the war would have been over," said Bloch, "by the time the machine would tackle it at a higher degree" of accuracy (Beyer *Information Age* 116). In this regard, Hopper's knowledge of round-off and truncation error, learnt from a chemistry course she audited at Vassar, served the team especially well. Her work on partial differential equations with Richard Courant was also valuable here (Beyer Information Age 54, 55, 116). Because Problem K was a closely-guarded military secret, detailed discussion of the results are not possible; as it is, such mathematics are beyond the scope of this thesis. However, one can clearly see that saving calculating time was an urgent priority, one stressed throughout the computation lab's processes and Grace's manual (e.g., Hopper et al. Mark I Manual 22, 27, 97). One major reason for this was the limited processing speed of the computer itself.

Programming: Speed

Though significantly slower than ENIAC, the Mark I ASCC and her crew boasted an uptime of nearly a hundred percent: "for the entirety of the war, the Mark I was operating 24 hours a day, 7 days a week, with the staff working three 8-hour shifts" (Beyer *Information Age* 90). The machine also represented a leap forward over human computers, mathematicians armed with desktop adding machines (Williams 30). According to Hopper, given that a human computer could work for about six hours before fatigue caused "a prohibitive number of errors," the Mark I ASCC ran "well nigh 100 times as fast," and, based on a 24-hour schedule, could complete almost six month's work "in a single day" (*Mark I Manual* 51).

With a cycle time of 300 microseconds, the Mark I ASCC completed 3 instructions per second (Beyer *Information Age* 7). Simple problems such as addition and subtraction took 6 seconds or less; more complex operations, such as finding the sine of an angle, took up to 90 seconds for one argument (Hopper et al. Mark I Manual 51). Multiplication and division, which required separate specialized hardware that had to be plugged into the main computer, took much longer; division wasted so much computing time that Hopper recommended using multiplication to solve for reciprocals instead (*Mark I Manual* 27). Operators also had to decide beforehand whether they would multiply or divide, because each operation used different units that could not be plugged in at the same time (Hopper et al. *Mark I Manual* 21).

The manual is thus rife with suggestions for saving computing time, devoting a section to "interposition," a crude form of multitasking (Hopper et al. *Mark I Manual* 98). In many longer operations, such as division or an interpolation of a series of functions, additional steps could be inserted into the problem: "addition, reset, reading into a print[er] counter or any other operation not involving the multiply-divide unit or the interpolators" could be inserted into the last line of code, and the typewriters could be turned on or off at the line before that—prepping the printer before the computer began checking its answers (Hopper et al. *Mark I Manual* 186). Considering that the completion of a large table of functions could take hours or days, every step interposed this way "easily translate[d] into weeks of saved run time" (Beyer *Information Age* 63).

The programmers also often used pragmatism and mathematical foreknowledge to develop time-saving solutions on the front end. Many problems with real-world applications did not require a full twenty-three decimal place calculation—Bloch discovered that the pitch and roll of a battleship, for example, could easily be expressed with values to four places (Beyer *Information Age* 55). Hopper, meanwhile, designed and implemented interposed instructions that made the computer automatically add page numbers to its printouts (Beyer *Information Age* 60). Initially, Aiken opposed the idea that the computer spend precious cycles doing anything other than "makin' numbers," as he liked to call their equations, but when Hopper pointed out the significant time savings in error-checking and program retrieval—ie, programmers could now recall exactly which page or card errors were on—he relented (Beyer *Information Age* 60). This seemingly minor modification also assisted with program review and storage: in a world where software was stacks of punched paper, programmers could now consult an exact code on a specific page; it also added a certain automated, professional polish to what had before been difficult to re-insert, typeset, and accurately paginate by hand (Beyer *Information Age* 60).

Programming: Punched Tape and Punched Cards

Because they are no longer used, it is important to briefly explain punched tape and punched cards: often traced back to the development of Jacquard loom cards, Mark I ASCC punched cards used existing IBM designs based on those invented by Herman Hollerith in 1889 (Hopper et al. *Mark I Manual* 5, 95; Hopper and Aiken *ASCC I* 385). Punched cards and punched tape served distinct software roles: the "sequence tape" or "control tape," like the paper tape on a player piano, ran all programs and signaled the computer to start and stop (Hopper et al. *Mark I Manual* 11). Punched cards held values or variables that were too big for the machine to calculate all at once, or signaled the computer to accept a different sequence tape, and could also be used to interpose certain values or commands (Hopper et al. *Mark I Manual* 42). It was vital that both tape and cards be inserted in proper forward order: the computer could not reverse itself to repeat commands or numbers (Beyer *Information Age* 62; Aiken and Hopper *ASCC 3* 522). Instead, any program with multiple repeated steps—such as differential equations requiring many small additions—meant coding in each operation one-by-one.

Mark I ASCC sequence tapes were 3-inch-wide paper ribbons with a series of sixteen holes per row in three separate groups of eight (Beyer *Information Age* 47). The first column, "IN," told the computer where to find the stored value it was looking for, and the second, "OUT," told it where to place answers (Hopper et al. *Mark I Manual* 12). The third column, "MISC," was reserved for "operational codes" such as addition, movement of values into or out of storage, or continuing an operation to the next line (Beyer *Information Age* 47; Hopper et al. *Mark I Manual* 12). To complicate matters, there were automatic and non-automatic codes; the computer would stop after a non-automatic code unless operators intervened (Hopper et al. *Mark I Manual* 99). Numerical values, meanwhile, could be read horizontally with enough practice: "the number '753' was represented by holes in the 7, 5, and 3 places on the tape," with the number 9 representing a minus sign (Beyer *Information Age* 47). There was no sign for addition because the storage counters did not distinguish between summing numbers and storing them, and simply assumed all values were positive unless prompted otherwise (Hopper et al. *Mark I Manual* 15, 16).

From this chapter, readers should take away an idea of the grand scale and physical complexity of America's first digital computer. It should also now be clear that while it was thick with complex "control information" many modern technical communicators strive to avoid, Hopper's manual for the Mark I ASCC served more than just its users' basic operating needs (Carroll et al. 127). Hopper's use of historical context to chart the development of the Mark I ASCC not only demystifies the computer itself, but rhetorically situates the computer as the natural and indeed almost inevitable product of grand ideas from great thinkers (Hopper et al. *Mark I Manual* 5, 6, 11). Her incorporation of these examples agrees with Todd's theory that grand ideas incite professional interest and personalize technology (47). Moreover, Hopper's consistent use of specific solutions to examples of real problems her users would face serves the audience's need for immediately-applicable information (Carroll et al. 151).

These factors make the *Manual of Operation's* strengths and weaknesses worthy of study in more detail. Chapter Four will illustrate ways Hopper's manual embodies particular criteria of Markel's measures of technical writing excellence, concluding with ways that Hopper could have used modern technical communication techniques to improve certain other dimensions of her writing, such as conciseness, accessibility, and improvements in error recovery or troubleshooting information (Markel 14;Carroll et al. 127).

CHAPTER FOUR: BEYOND THE EDUCATION OF A COMPUTER

Grace Hopper's *Manual of Operation* displays several strong features of good technical writing: in particular, Hopper does an excellent job of establishing adequate context for novel concepts, ensuring that her audience has sufficient background to understand and complete problems on their own. In addition, her information is always provided in a consistent way, proceeding in order of conceptual complexity. This consistency drives home the core strength of the Mark I manual: Hopper's thorough detail and attention to completeness. Thus, one might say Hopper's core strengths as a technical communicator are context, consistency, and completeness.

One useful metric for these features comes from Markel (2012), who holds that there are eight distinct qualities present in excellent technical writing (13, 14). As this chapter will illustrate, the *Manual of Operation* embodies at least four of these criteria: comprehensiveness, correctness, clarity, and accuracy. The manual also prominently features Markel's dimensions of honesty and professional appearance, each of which are best represented as functions of Grace's comprehensiveness and accuracy. (That is, this study will not address honesty and professional appearance at length, but will treat them as part of the other four major strengths her manual exhibits.) Ways that modern technical communication techniques could improve Hopper's manual, such as the use of minimalist principles, information mapping, and topic complexity, will also be discussed (Carroll, et al.; Ganier; Karreman and Steehouder).

This chapter will also introduce Hopper's paper "The Education of a Computer," a lucid discussion of UNIVAC architecture and the first documentation ever written about software compilers (Mitchell 2). "The Education of a Computer" not only showcases Hopper's stylistic growth and improved command of conciseness, but also offers solid insights for today's technical communication professionals in terms of Hopper's practical, results-driven approach and clear concern with audience needs.

The Manual of Operation

Weighing in at 561 pages, the *Manual of Operation* was the antithesis of light bedside reading (Williams 64). Part codebook, part reference manual, and part public relations attempt, this massive manual was formally credited to the Harvard Computation Lab, because its construction represented a team effort (Hopper et al. *Mark I Manual Preface,* n.p.; Beyer *Information Age* 125). Grace Hopper, however, is the sole author of more than half its contents, and according to Aiken himself, "more than any other person is responsible for completion of the book" (Hopper et al. *Mark I Manual Preface,* n.p.). Except for some of the example problems in Chapter VI, co-authored with Brooks J. Lockheart, and the codebook, compiled based on shared work with fellow programmers Bloch and Campbell, the lion's share of the writing is hers (Hopper et al. *Mark I Manual Preface,* n.p.).

Manual of Operation: Organization and Presentation

The Preface of the manual distributes writing credit amongst the members of the Harvard Computation Lab team: Eunice McMasters for the diagrams, Harry Goheen for the bibliography, and Brooks J. Lockhart for majority contribution to Chapter VI (Solution of Examples); the Preface also names Grace Hopper the foremost author of the documentation, lead editor of the volume, and a main author of Chapters IV (Coding) and V (Plugging Instructions) (Hopper et al. *Mark I Manual Preface,* n.p.). The main table of contents is the manual's central point of reference; all major asides, such as Hopper's recommendation to consult "a full discussion of all codes" to locate specific operations, are given as full chapter names: "in Chapter IV, Coding," without inclusion of specific page numbers (Hopper et al. *Mark I Manual Contents*, n.p.; 14). The List of Plates (photographs) and List of Figures (diagrams) function as another table of contents; however, while the photographs and diagrams themselves are clearly captioned, none of the captions refer back to the lists of Plates and Figures, making it difficult to locate specific charts, particularly after a protracted search or long reading (e.g. Hopper et al. *Mark I Manual* III; Markel 309, 446). One likely reason for this discontinuity is that front and back matter are often some of the final work completed on a manual, and like many technical communicators, Hopper was writing on a tight timeline and tighter budget, without direct control of the document's design (Markel 531; Hackos 404, 561).

Chapter organization of the Mark I manual is straightforward, beginning with a history of the computer's conceptual development in Chapter I. Hopper's writing truly shines here: beginning with the abacus, she connects the development of computers from the lofty "ideas of the physicists and mathematicians" to the increasing economic complexity and trade realities of the seventeenth century (Hopper et al. *Mark I Manual* 4, 5). In Hopper's account, Napier's bones are ideologically interconnected with the pragmatics of Pascal's adding machines, which are themselves seen as the ancestors of Charles Babbage's analytical and difference engines (Hopper et al. *Mark I Manual* 5, 6). This is an important point: in her study of Babbage's work, Grace discovered the research of Ada Byron Lovelace, who wrote the first loop, and who in turn was an inspiration for Hopper's invention of the compiler in 1952 (Williams 82). For his own part, Aiken had long been aware of the reputational cachet inherent in the Mark I ASCC's ideological ancestry, and had deliberately cultivated an association between his machine and Babbage's for years; as early as 1938, Aiken systematically cited Babbage as his hero (Beyer *Information Age* 130, 136). Aiken also corresponded with Babbage's grandson, Richard, who was glad of the recognition for his ancestor and of the interpersonal connection to noteworthy scientific ideas (131, 136).

Chapter II, Description of the Machine, itemizes the Mark I's components in intensive detail, from the register switches up, and Chapter III, Electrical Circuits, describes how the computer solved a problem, cycle-by-cycle, in terms of the wiring itself (Hopper et al. *Mark I Manual* 10; 80 - 95). Chapter III is particularly helpful to technical communicators decoding extinct machinery entirely from its documentation. It provides detailed images of a fully deconstructed relay and disassembled counter, with component parts carefully labeled (Hopper et al. *Mark I Manual* XVI, XVII). Similarly, while Hopper's cycle-by-cycle elucidation of processor function is top-heavy by modern standards, containing such lexical whoppers as "The amount of shift combined in a storage counter with a constant dependent upon the position of the operating decimal point supplies the exponent required. Further examples of special controls associated with the multiply unit will be described later in connection with the discussion of the electro-mechanical tables of the elementary transcendental functions," such thick description was necessary for electrical engineers and machine operators who were intimately familiar with electromechanical relay technology, but who had never used or even seen a computer before (Hopper et al. *Mark I Manual* 24).

The title for Chapter IV, Coding, is somewhat misleading: from the first pages of Chapter I, Grace discusses mathematics in terms of how the computer 'thinks,' explaining that with end-around carry, the computer solves addition in two steps by performing all carrying at once, rather than per-column as a human would (Hopper et al. *Mark I Manual* 15). Because programming is the end goal of Hopper's intended audience, each chapter of the manual furthers that objective in some way.

Hopper's organizational schema proceeds in order of conceptual complexity: when discussing operations, she always begins by explaining how the Mark I ASCC completes an addition—except in Chapter IV, Coding, where she starts with a multiplication example (Hopper et al. *Mark I Manual* 105, 111). This is both because of her audience's presumed knowledge by this point, and because multiplication was a more complex task, requiring additional manipulation by plugboard, and therefore more detailed explanation (Hopper et al. *Mark I Manual* 111). Chapter V, Plugging Instructions, dealt with the separate needs of the plugboard architecture, and Chapter VI, Solution of Examples, assisted users in writing their own programs (Hopper et al. *Mark I Manual* 245, 287). The Appendices, meanwhile, were exhaustive references, filled with material largely of interest to technicians and operators rather than coders; they comprised a full list of the switches, circuits, diagrams, and so on (Hopper et al. *Mark I Manual Contents*, n.p.; Rubens 48).

Comprehensiveness and Correctness

In terms of Markel's dimensions of technical writing excellence, comprehensiveness is the balancing act of leaving the user with exactly as much information as they need to understand concepts and complete tasks. While this may seem self-evident, it is in fact complex: good user manuals must balance the needs of both novice and experienced users, who will approach the text differently and with distinct requirements (Ganier 15). Many researchers, for example, distinguish between declarative information and procedural information in instructions (Karreman and Steehouder 34). Declarative information is background data about the equipment or software, such as menu descriptions, whereas procedural information shows how to complete tasks or perform actions. However, the specific type of information—whether procedural or declarative—is less important for comprehensiveness: rather, the thoroughness of the information is what matters. Modern scholars encourage an emphasis on task-oriented procedural information that focuses on users' specific end goals (Carroll et al.; Ganier; Markel). Markel argues that "comprehensiveness is crucial because readers need complete, self-contained discussion in order to use the information safely, effectively, and efficiently," and moreover that manuals—particularly large reference manuals, such as Hopper's— "often serve as the official company record of a project, from its inception to its completion" (14).

For Markel, correctness is chiefly grammatical: "conventions of grammar, punctuation, spelling, mechanics, and usage," which can be thought of as the avoidance of lexical and syntax error (14). Although the grammar of Hopper's manual can best be described as top-heavy, it is also scrupulously correct, with nary a clause lost among the dense compound sentences that detail, for example, cycle time. Moreover, Hopper's writing unerringly follows grammatical conventions of the day, such as including phrasal commas that most present-day business communication omits: consider that "In 1617, John Napier, following his invention of logarithms, published an account of his numbering rods, known as 'Napier's Bones'" (Hopper et al. Mark I Manual 1). For speed and clarity, modern technical writers might be encouraged to structure the sentence thusly: "Following his invention of logarithms, John Napier published an account of his 'Napier's Bones' numbering rods in 1617," removing four of five stopping points while preserving each central idea and active clause (for example, Einsohn). However, Hopper's original reading imparts a certain grandeur to the sentence, and to the great thinker and important invention at its heart; similar stately sentences are woven together to achieve an impressive and even intimidating long-form manual rich with detailed technical data. Hopper's strict adherence to Markel's dimensions of comprehensiveness and correctness thus creates and reinforces a perception of a profoundly professional document detailing a highly technical piece of equipment: in this way, the entire presentation of Hopper's manual functions as an appeal to ethos.

Honesty, Clarity, and Accuracy

For Hopper and the staff of the Harvard Computation Lab, Markel's call to honesty in technical documents would have been redundant: the purpose of the document was to get the computer working as well and as fast as possible and create maximum uptime while avoiding any problems. Because of the urgent nature of their calculations—munitions data for active battlefield situations, optimal submarine deployment, rocket trajectories–and their intent for immediate use, there was no room for deception in the instructions. Civilian applications of the technology were flatly not a concern; therefore, emotional appeals in the manual were largely limited to the history section. While certain sections are relentlessly optimistic–consider Hopper's promise of "six months' work in a single day" – this statement itself assumes the machine is working perfectly, and does not represent a deliberate attempt to defraud anyone. Similarly, the manual is very honest about what the machine cannot do, such as its inability to rewind an instruction (Aiken and Hopper *ASCC 3* 522).

For Markel, accuracy is a question of tone: excellent technical writing should be objective, with a factual presentation and a lack of "spin" (13). Markel stresses that it is vitally important that an accurate document be as "as objective and unbiased" as possible (13). The *Manual of Operation* fits these criteria quite well. Aside from the first chapter, which establishes Aiken as a great thinker with an amazing invention, the rest of the manual is crisply factual and almost painfully dry. Hopper does not hesitate to point out problems and peculiarities in the hardware: "sines of third and fourth quadrant angles cannot be computed directly [at] eleven or fewer operating decimal places" (182). Square roots could not be processed directly, and required "two divisions, one multiplication, and four additions," or an even more intensive "two multiplications and two divisions," just to reach the first approximation needed to *begin* isolating the answer—a set of operations that would have to be repeated until the root was found, gobbling processing power and time (181). "The sign counter [of the multiply/divide hardware] is the only one in the machine which cannot be reset by button" (72). These explanations are given as bare fact, without hedging or euphemism, and with alternate options where possible (as with the square root example; Hopper et al. *Mark I Manual* 182). Accuracy, clarity, and utility were the overriding focus of Hopper's manual, designed to be read by a specialized audience for highly technical purposes during wartime

Clarity, or conveying "a single meaning the reader can understand easily," is another strong feature of Hopper's manual (Markel 13). While the technical specifications of the ASCC are sometimes smothered in extraneous grammar, Hopper is particularly good at explaining unfamiliar concepts in terms of known technology; she discusses at length the components and interlocking nature of the counter flywheels, and breaks down in detail the exact process by which the central crankshaft powers the computer (for example, Hopper et al. *Mark I Manual* 59). Ganier (2004) recommends the use of coordinated, "mixed" text and graphics to maximize clarity when explaining technical information or machinery, which is in abundant evidence here—nearly a third of Chapters II and III are devoted to diagrams and photo plates (see, e.g., Hopper et al. *Mark I Manual* 18, 19, 42, 44, 57, 61; Ganier 20). Ganier also recommends labeling to enhance comprehension, and though it is cumbersome here, Hopper does use labeling to point out, for example, which relays are engaged to run the start and stop circuits for one cycle (Ganier 20, 21; Hopper et al. *Mark I Manual* 57). Hopper devotes the entirety of Chapter III to the layout of the circuitry itself; these diagrams and explanations would be instantly recognizable to electrical engineers of the day. When wading through such dense, detailed passages—which acolytes of the modern minimalist approach might consider the opposite of clarity—it is important to remember that for Hopper's original audience, heavy expository copy was standard for technical specifications, and moreover was necessary in order to explain a previously unheard-of piece of technology: the digital computer.

Room for Improvement

In Markel's model, there are three more dimensions of excellent technical writing: conciseness, clarity, and accessibility. These were Hopper's challenge areas, and conciseness in particular was a major weakness of the manual. This is obvious in comparison with the task-oriented minimalism favored by modern technical communication researchers, such as Carroll and colleagues (1987). Though Hopper's audience were neophytes who had never seen a computer before, one can assume that they would swiftly learn, yet still be bogged down by the linear, step-by-step approach she uses. Ganier (2004) notes that this is a particular problem for all reference manuals (15).

Conciseness is more than the art of brevity: to be effective, leaner documents must still "be useful to a busy reader" (Markel 14). Flabby prose can introduce unintended meanings which may in turn lead to costly technical errors (Markel 14). Moreover, producing well-written, compact, error-free technical documents reflects positively on the technical communicator as a writer and as a subject matter expert, and on their organization at large; conciseness thus confers a powerful, professional ethos (228).

In addition to such familiar truisms as removing unnecessary phrases, choosing shorter words, and using the active voice, Markel offers several additional specific suggestions to improve conciseness, including: (1) use lists, (2) emphasize new and important information, (3) choose appropriate sentence lengths, (4) focus on the subject (5) focus on the verb, (6) improve parallelism, and (7) use modifiers effectively (14, 228).

Not originally presented as a list, these suggestions were provided as one to illustrate Markel's observation that hefty sentences may prevent readers from concentrating on new information. With lists, audiences can literally "see how many phrases they have to remember": there is less cognitive load with new information when complex clauses are truncated into compact lists (Markel 229).

Another of Markel's exhortations is to be specific: whether writing about "an automobile, a rail, or a can of tomatoes," use the best possible noun for it—while calling it a Ford Focus may be undesirable because it might be taken for endorsement, "the car" is more specific than "the vehicle," and much clearer than "the thing" (Hopper "Education" 243; Markel 243). To increase specificity, use positive construction: "most" instead of "not all." Saying "on schedule" is more powerful than "not late." (Markel 244; Ganier 18). Moreover, arrangement "of the words on the page" reinforces and enhances meaning, especially when dealing with new information--"sentences are often easier to understand and more emphatic if new information appears at the end" (Markel 229, 230, 231).

When writing the *Manual of Operation,* Grace Hopper would have benefited enormously from the power of lists--a convention she almost exclusively reserved for diagrams (Hopper et al. *Mark I Manual* 62, 78, 99). Their use is most notable in Chapter III, Electrical Circuits, in which Hopper explains the ASCC's central mechanics: her Figure 22, the Mechanical Drive System, is accompanied by letter labeling (for major components A – K) and described in full-prose detail over the next two and a half pages (Hopper et al. *Mark I Manual* 58 – 60). Even her abbreviated description written with Aiken for *Electrical Engineering* is daunting and dense: "The sequence and interpolator mechanisms and the counter wheels are all driven by a single gear-connected mechanical system […in] Figure 4, A is a line shaft extending nearly the full length of the calculator" and "driven by the 5-horsepower motor, B. This shaft is contained in the shaft housing shown near the base of the machine in Figures 2 and 3. The main sequence mechanism and the three interpolator mechanisms are supplied with mechanical power by the roller sprocket drives, C and D respectively. The spiral gears, E, connect to the main drive shafts, F. These in turn are connected to the horizontal shafts, G, through the spiral gears, H. On the shaft, G, are mounted 12 or fewer gear wheels (J of figure 4) each of which supplies mechanical power to a single counter wheel by engaging with the gear shown in the partially assembled counter." (Aiken and Hopper ASCC I 386, 387). To observe the original diagram of the ASCC mechanical drive system in context, see Figure 1 on the following page.

58

ELECTRICAL CIRCUITS

concerned with the sequences of operations necessary to the control of the functional units. However, it is possible to construct a subsidiary sequence control for any given purpose. For example, the evaluation of a definite integral may be reduced to the computation of values of the integrand for equidistant values of the argument by a short control tape, which also directs a subsidiary sequence control wired to apply a general quadrature formula. In this instance, the coding necessary to the evaluation of definite integrals is greatly reduced. Such specialized subsidiary sequence controls are added to the calculator from time to time as may be desired. These differ only in the sense that some control a greater number of operations and in that their control extends over a longer period of time. Unfortunately, space will not permit the description of all of the sequence controls in the calculator. The fact that those controlling multiplication and division are not only the most simple, but also the more basic in computation, dictates their choice for detailed discussion.

Before entering upon this subject, however, it will be necessary to discuss the use of counters and their drive. Referring to Fig. 22, A is a line shaft extending nearly the full length of the calculator and driven by the four horsepower motor, B. This shaft is contained in the shaft housing shown near the base of the machine in Plates II, III and XII. The main sequence mechanism and the three interpolator mechanisms are supplied with mechanical power by roller chain and sprocket drives, C and D,

MECHANICAL DRIVE SYSTEM

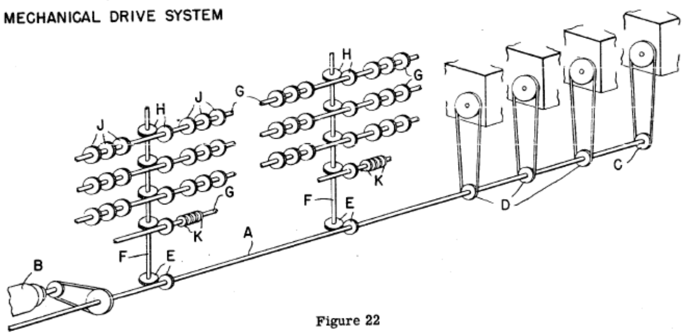
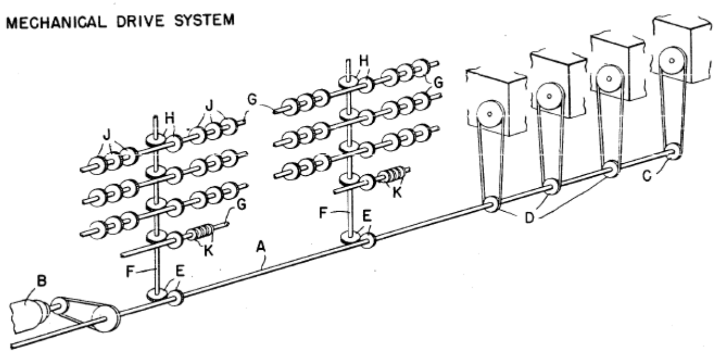


Figure 22

Figure 1. Diagram of the Mark I ASCC mechanical drive system, shown with original text.
("Mechanical Drive System" in original, as shown.)
Source: Hopper et al. *A Manual of Operation for the Automatic Sequence Controlled Calculator*, pp. 58.

Surely, a single "mixed" text and graphics diagram as recommended by Ganier would be a much more concise and equally fluent alternative (18).

See Figure 2 on the following page for an idea of how this might be accomplished.



A: main drive shaft	
B: main motor (5 horsepower)	
C, D: roller chain and sprocket drives (propulsion)	
E: spiral gears (connect A and F)	
F: main vertical shafts	
G: main horizontal shafts	
H: spiral gears (connect F and G)	
J: gear wheels (to storage counters)	

Figure 2. Proposed edited reproduction of diagram for Harvard Mark I ASCC mechanical drive system.
Image Source: Hopper et al., *A Manual of Operation for the Automatic Sequence Controlled Calculator*, pp. 58.
(Diagram format and presentation are student's own work.)

The elimination of nonrestrictive clauses could also have shortened the diagram, streamlining Hopper's information in preparation for a list (237, 238). (Recall that nonrestrictive clauses, such as this phrase in this sentence, can be removed without impacting either the structure or the informational value of the sentence itself. Without its nonrestrictive clause, this sentence reads, "Recall that nonrestrictive clauses can be removed without impacting either the structure or the informational value of the sentence itself.")

Moreover, even without the use of lists, and even with all her clauses intact, if Hopper had followed Markel's suggestion to limit prepositional phrases—references to items "on" or "under" other items—these sections might have gained both greater clarity and conciseness (Markel 247). Consider, for example, that "the shaft housing down near the base of the machine" could be called "main driveshaft housing," which is shorter and— in combination with the diagram—specifies a precise location (Hopper et al. *Mark I Manual* 58). As a final suggestion to improve conciseness, Markel recommends using "your software to compute the average sentence length of a representative passage," and using that data to edit for length based on your audience's needs (233). I wonder what Grace would think of that.

Carroll and colleagues (1986) likewise offer several suggestions to improve conciseness. Echoing Markel's suggestions to condense information into lists and use positive construction to clarify commands, Carroll et al. further recommend referring to new hardware, procedures, or peripherals concretely: "display monitors" are monitors, and "the system" or "the computer" should be called by its proper name (for example, "the ASCC" or " UNIVAC") (127, 131). Hopper agrees with this advice in "The Education of a Computer": "I shall use UNIVAC as synonymous with electronic digital computer; primarily because I think that way, but also because it is convenient" (243).

Carroll and colleagues not only found that that conciseness promotes comprehension and performance among learners new to the task, system, or software; but they also focused on a distinct but related dimension of excellent technical writing: accessibility (Markel 14, Carroll et al. 127). Accessibility is an interesting criterion, because it is a new concern; before and even during the 1980s, large print manuals crowded with declarative information were the norm (Carroll et al. 127).

Lengthy, complex sentences were common to the military manuals of Grace's era. For example, consider a technical manual supplement for gear with a comparable purpose, audience, and time-frame: a 1944 United States National Defense Research Commission Army/Navy joint brief on sonar standards and equipment (USNDRC, "Calibration Measurements" 1944). The sonar brief contains this elegant but overbearing purpose statement: "One of the most important requirements for a standard is dependability, that is, a standard should give the same performance day in and day out under various conditions of use so that, once its calibration has been determined, it can be relied upon for an extensive period of time and under a wide range of testing conditions" (USNDRC "Calibration Measurements" 2). Even pointed recommendations are given in passive voice and bracketed with phrasal commas: "The X-cut Rochelle salt crystal, because of its temperature impedance, for instance, should in general be avoided for standards" (USNDRC "Calibration Measurements" 2). Modern technical communicators would doubtless include, "Tip: Avoid using X-cut Rochelle salt crystals in the creation of standard devices because of high temperature impedance." Such a tip could be further enhanced with graphics, color, or headings to mark it as distinct from the body text—features not used in the sonar supplement (USNDRC "Calibration Measurements" 2; Ganier 20; Markel 206, 207). While none of these suggested features are present in the brief itself, the use of some or all of them would improve both comprehensiveness and accessibility. In short, Grace's *Manual of Operation* is comparable in tone and style to the works of her contemporaries, which show similar areas for improvement.

Moreover, though Grace was the lead writer and editor for the *Manual of Operation*, its construction was a team effort—something still common to reference manuals today. Because most reference manuals are comprised of many smaller sections, often written by different writers, they are typically compiled out of sequence (Hackos 404). Details may be repeated between sections, along with added declarative “control information” intended to orient the user, which further lengthens and complicates the manual (Carroll et al. 127). Given these structural limitations of reference manuals, it is imperative that readers be able to locate their desired information when they need it: this is accessibility (Markel 14).

Carroll and colleagues found that instructions highlighting specific step-by-step tasks, offset by clear headings, reduced cognitive load for new users and decreased their “starting up” time when using a new technology or learning a new skill (128). Users given a minimalist manual with specific task instructions learned new tasks almost 50% faster than users given a comparable long, reference-style self-instruction manual (Carroll et al. 140). In a follow-up experiment, Carroll and Mazur determined that including error recovery (troubleshooting) information improved performance in the minimal manual group over the traditional manual group (145, 146). Moreover, because minimal manual users could find error recovery information easily in the instructions, they spent 20% less time than traditional users relying on the external system library for additional help (Carroll et al.146). Minimal manual users spent an average of 2 minutes seeking external help, versus almost 16 minutes for the long manual group (Carroll et al.146). Accessibility makes a clear difference for both task completion and error recovery.

To increase accessibility, Carroll et al. recommend first drafting to concrete tasks users will want to accomplish, then creating task-oriented procedures showcasing these specific tasks (129, 131). Keeping the audience’s task needs in mind increases readability, accessibility, and time spent learning the tasks, particularly at introductory levels (Carroll et al. 146).

Markel has several concrete suggestions that also improve accessibility: 1) write coherent titles to give readers their "first clue" whether they can find their needed information; 2) use clear headings to "communicate the relative importance" of ideas and convey their overall topical relationship; 3) use lists; and 4) use clear relationships between information (205, 207, 213). For example, consider the importance of consistency when emphasizing information: are important data "set off by headings" or sectioned off into clear lists (Markel 213)? Clear hierarchy and structure of ideas spares the audience flipping back and forth between sections, because they learn when and where to consistently expect certain types of information (Markel 14, 211). Ganier agrees that in instructional situations, clear headings both "facilitate the location of information" and "the comprehension and execution of instructions" (20). By following the minimalist principles of modern technical communicators, Hopper could have improved both accessibility and conciseness at one stroke.

Certainly, accessibility and conciseness are two major areas in which Hopper could take a lesson from today's technical communicators, and not the other way around. However, one of her best-known later works, "The Education of a Computer," showcases ways in which Hopper later used many modern principles effectively (Hopper "Education" 243, 247, 248). Though no instructions are perfect, some still offer valuable lessons to today's technical communicators, and "The Education of a Computer" is both a minor classic in computing literature and a definitive example of Hopper's growth and development as a technical writer.

“The Education of a Computer,” Electrical Engineering, and the Evolution of a Technical Writer

> “It is the current aim to replace, as far as possible, the human brain by an electronic digital computer.” –Grace Hopper

Hopper’s article on UNIVAC architecture for the Remington Rand Corporation is not only the first documentation ever written concerning compilers, nor is it merely a showcase for the evolution of Grace’s writing style and approach to material. It is also a demonstration that, as far as possible, Hopper began at the beginning and always started with what played best to the house: a brief and flashy historical overview charting the grand ideas of great inventors, tracing “the mechanization of mechanical thinking” from the abacus through Pascal, Leibenitz, and Babbage; culminating in the work of Professor Howard H. Aiken of Harvard University, Dr. John W Mauchly of Eckert-Mauchly, and Dr. M. V. Wilkes of the University of Cambridge (Hopper *Education* 243; Beyer 222). She credits Aiken specifically with the “idea of a library of routines described in the Mark I manual,” Dr. Mauchly with “the basic principles of the ‘short-order code’ and suggestions, criticism, and untiring patience” as informal editor and advisor; and “from Dr. Wilkes, the greatest help of all, a book on the subject,” for all of which Hopper most earnestly expresses her debt and appreciation (Hopper *Education* 243). Grace condensed a fifteen-page history of great thinkers into a compact, powerful, half-page emotional appeal that establishes her mentors as formidable intellects—and herself, by extension, as masterful and competent, possessed of expert knowledge that is worthwhile and well-vetted.

“The Education of a Computer” also showcases the evolution of Grace’s writing style; she begins with graphical information this time, using Figure 2, below (also shown as Figure 2 in original document) to illustrate the minimum resources a person needs to solve a math problem (243).

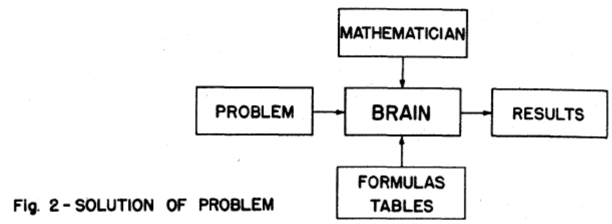


Figure 3. Minimum resources required for mental solution of a math problem.
("Solution of Problem" in original, as shown.)
Source: Grace Hopper, "The Education of a Computer" pp. 243.

Hopper argues that even mental math is like programming, involving "input to the operations; controls, even if they be only start and stop; previously prepared tools to supply data to the operation; and output of products, which may, in turn, become the input of another operation" (Hopper *Education* 243, 244). She further compares programming to a production line: using "raw materials, controlled by human beings, possibly through instruments, supplied with machine tools, the operation produces an automobile, a rail, or a can of tomatoes" (Hopper *Education* 243).

Arguing that all present users of UNIVAC—"armed services, government, and industry"—want not just to create new operations but also to improve existing ones, Hopper contends that the easiest way to do so is to offload as much menial calculation to computers as possible. Short of a robot uprising, this is what she means by replacing a human brain (Hopper *Education* 243). See Figure 3, following, for the full picture of human-computer interaction in Hopper's model.

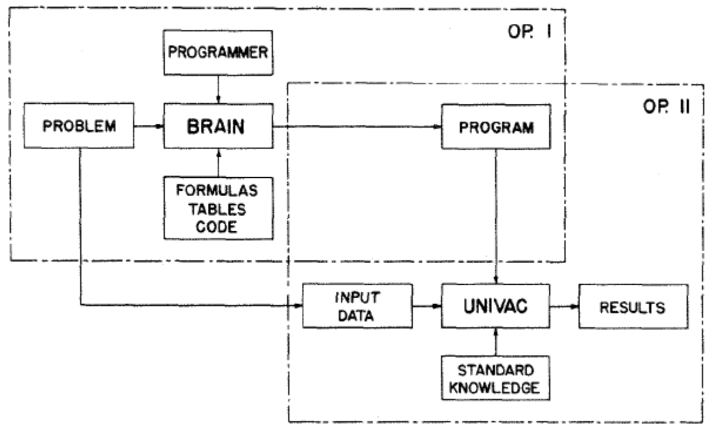


Figure 4. Minimum resources required for solution of a math problem using human-computer interaction.
Source: Grace Hopper, "The Education of a Computer" pp. 244

By the time a problem reaches Figure 3 (corresponding to Figure 4 in the article), the mathematician is now *de facto* a programmer and supplies code to the computer that turns the component parts of equations into UNIVAC-readable data—and so long as the novelty of programming holds, this solution is sufficient (Hopper *Education* 243). However, once the thrill of creating code becomes the tedium of troubleshooting it, that intensive, effort-laden process "now looms as an imposition on the human brain," and with the computer paid for, the added costs of programming (and consumed computing labor and time) attract the notice of "vice-presidents and project directors" (Hopper *Education* 243). Starting with human visualization of the problem, Hopper adds the UNIVAC, which draws on the instructions input by the mathematician "under the control of 'a compiling routine of Type A', using subroutines and its own instruction code" to produce a program, which then calculates the inputted data (*Education* 244). By transferring all raw calculating responsibilities to the UNIVAC, mathematicians gain "a major reduction in time consumed and in sources of error" (*Education* 244).

Compilers and subroutines were a vital development in software, and are the foundation of the current computing paradigm (Beyer 10; 16; 223). For Hopper's main audience—other mathematicians—the chief value of this new technique lay in crunching larger values faster, enabling exciting discoveries at the apparent press of a button. Armed with a catalog of subroutines, the "programmer may return to being a mathematician," who no longer strictly needs to carry bulky indexes of t-tables around; "[he] does not even need to know the particular instruction code used by the computer," instead requiring working knowledge of how to use UNIVAC's subroutine catalog (Hopper *Education* 243, 244). Mathematicians fed UNIVAC data and subroutines from the catalog, which it then used to fetch those "subroutines and its own instruction code" to compile a program that ran the mathematician's problem more or less automatically (Hopper *Education* 244). Grace emphasizes that if the available subroutine library is a good one, programming time has "been reduced to a matter of hours, rather than weeks" (Hopper *Education* 244). Stressing ease of use, improved results, and the capacity to solve problems absent the tedium of coding them manually, Hopper makes a vivid case for the importance of subroutines in programming, and does so in terms that matter to her core audience. Grace has also tailored her tone throughout to be friendlier and more commercial while maintaining professionalism; "[in] any event, the mathematician need only state 'go to operation k,' and the compiling routine does the rest" ("Education" 246).

With "The Education of a Computer," Hopper has taken Carroll and colleagues' admonition to focus on users' goals and tasks to heart. Hopper aligns her examples with tasks the audience wants to perform: supposing "the mathematician wishes to evaluate a function and its first *n* derivatives," Hopper outlays how to tabulate the needed values in UNIVAC-readable form, using her expertise as a mathematician to appeal directly to fellow mathematicians in a concrete way (“Education” 249). In line with the recommendations of Carroll et al., Hopper has provided minimal, abbreviated orienting information, such as “[p]resuming that code, program, input data, and results are familiar terms,” before discussing more specialized data, such as how to specify “the forms of information and routines acceptable to the system,” (Hopper “Education” 244). From there, defining what a subroutine is and how it works, and explaining the difference between compiling routines – Type A and B—and task routines all flow quite naturally, in order of complexity (Hopper “Education” 245, 248). Here, Hopper’s emphasis is clearly on conciseness and accessibility.

Considered with her work on the Mark I ASCC manual, “The Education of a Computer” teaches us several things about Hopper’s approach to her audience and her subject material, from which modern technical communicators can learn a great deal. First, and foremost: Hopper consistently contextualizes issues and instructions in terms that matter to her audience. When addressing engineers, she explains the exigency of computers in mechanical terms: “[t]he increased accuracy of physical measurement has made necessary more accurate computation,” which computers provide (Aiken and Hopper *ASCC I* 386). When addressing mathematicians, she explains the necessity of Chapter III’s extensive mechanical details in terms of their future goals: “adequate preparation for the coding and plugging procedures to be discussed in the following two chapters […which] followed by a study of the examples […] will enable a mathematician to make full use of the calculator, and to exploit its facilities to the greatest possible advantage” (Hopper et al. *Mark I Manual* 97). By focusing on her audience’s goals and tasks, then illustrating how much more easily they can achieve the desired results with computers, Hopper has created not only strong persuasive appeals but also effective examples that help users understand and perform real tasks (e.g. Carroll et al. 128, 129).

Second, Hopper consistently highlights alternate uses for the information she teaches. When educating a computer, Hopper leverages her explanation of subroutines to show how *kernel* and *threading* subroutines can be used to automatically write more subroutines—the first major discussion of how to use a subroutine to compile code (Hopper "Education" 248, emphases added; Beyer 10). Hopper also stresses that, once mastered, subroutines can be used to "correct the computational procedure submitted"; effectively checking the both the computer's results and the mathematician's own work, then further highlights that a similar routine can be used to "supply estimates of running time with each program" (Hopper "Education" 249). The basic "87 stop" controls for the Mark I not only stop a line of coding, but can be used to interpose printing instructions, or to pass a partial solution to a separate plugged-in unit to finish the equation (Hopper *Mark I Manual* 99, 186). The solution for division—multiplying by reciprocals—is an alternate solution path to save computing time (Hopper *Mark I Manual* 21, 27). Hopper consistently offers her users more than one result from the same core concept.

Hopper's third and final major lesson for modern technical communicators is also arguably her strongest: when confronting new data, a new operational model, or a brand new way of doing something, she provides multiple examples and addresses them in order of complexity, always working from least to most. When instructing Mark I users, Hopper always begins with the same operation: addition (for example, *Mark I Manual* 15). In Chapter IV, Coding, Hopper begins with multiplication instead, giving users a stronger test of their mettle while also providing detailed instructions for the additional commands needed to connect and use the multiplying unit (*Mark I Manual* 105, 111). Explaining ASCC computing to electrical and mechanical engineers, Hopper starts with the overarching practical need to balance cycle-time savings with operational simplicity: the more complex the program, the more adjustments programmers had to perform on switches, counters, tapes, and plugboards, all of which could introduce operator error (Aiken and Hopper ASCC III 523). This general call for “meticulous precision” fed naturally into the necessity of operating instructions for every sequence control tape—an exigency that Hopper used, in turn, to explain how to prepare, encode, and check the results of a single, specific polynomial equation “selected for its mathematical simplicity” and covered in detail in the article (Aiken and Hopper ASCC 3 526). Hopper uses this entire section as an object lesson, offering step-by-step example operating instructions--and related plugging instructions--for the sample equation, so that engineers, operators, and attendants alike could run subsequent programs to solve equations on their own (Aiken and Hopper ASCC 3 526 - 528).

Grace Hopper’s technical writing has much to recommend it. Besides being the first major computer manual written in the United States, the *Manual of Operation* is a thorough and thoroughly grounded reference manual, replete with detailed historical and practical context, plentiful and exact example problems, and practical solutions to audience needs and goals (for example, Hopper et al. *Mark I Manual* 10, 15, 105, 111, 180). Her articles for *Electrical Engineering*, co-written with Aiken, discuss Mark I machinery in terms that matter to actual operators (Aiken and Hopper *ASCC I* 387, 388). The articles clarify certain concepts—such as the need for software planning documents—in plain language (Aiken and Hopper *ASCC I* 387; Aiken and Hopper *ASCC III* 325). While conciseness was the weakest feature of her work on the Mark I ASCC manual, Grace’s expansive style imparted a certain gravitas to the work that reflected the exigencies of wartime development and the need for precise, professional language when dealing with an entirely new technology (Hopper et al. *Mark I Manual* 1). Last, but certainly not least among Grace’s writings, “The Education of a Computer” is in some ways an inventor’s love letter; it discusses the need for a new sort of program and then elegantly explains exactly how compilers can be used to achieve it, as well as what the benefits are for mathematicians and organizations at large.

Examination of Hopper's work allows current technical communicators and future hopefuls a clear picture of the evolution of Grace's writing style while illustrating her command of comprehensiveness, correctness, clarity, and accuracy. She is a master at establishing adequate context, furnishing multiple complete examples that complete users' practical goals, and honestly and accurately addressing both the strengths and limitations of every device or concept she explains. Hopper never hesitates to be specific about what the computer can and cannot do—such as the Mark I's inability to rewind instructions, lack of a dedicated square root unit, limitations in calculating sine series, and special problems with printing or punching (Aiken and Hopper *ASCC 3* 523; Hopper *Mark I Manual* 170, 171, 182, 99). As for correctness, one need only review such intricate sentences as "The time interval necessary for the brush to traverse the between two successive [pins] is one-sixteenth of a cycle, the number spots being so spaced […] as to minimize the ratio of the mechanical backlash to the distance traversed between spots" to see that Hopper's grammar and style conferred a powerful and professional ethos to urgently needed documentation for important technological developments (Hopper et al. *Mark I Manual* 59).

CHAPTER FIVE: CONCLUSION

Technical communication is not a discipline one immediately associates with the idea of grand figures of history, science, or philosophy: though technical communication includes Chaucer, Benjamin Franklin, Herbert Hoover, and Hildegaard of Bingen among its luminaries, their work in the field is rarely discussed (for example, Todd, Malone, Rauch). While research interest in the history of technical communication has grown considerably in recent years, expanding into such areas as device schematics, shipwright's manuals, domestic handbooks, gardening practices almanacs or manuals, botany, and medical documentation, research into specific heroes remains to be done, and the role of notable women in technical communication remains under-researched, in particular (for example, Todd; Tebeaux and Moran; Tebeaux and Lay; Durack; Shirk; Rauch).

Because technical documentation written by women often concerns domestic devices (for example, sewing machines) or processes (for instance, gardening), it has often been disregarded in formal research (Rauch; Durack; Tebeaux). Given the dearth of research into notable technical communicators in general, and the lack of research into specific strong female voices in the field, locating and celebrating notable women technical communicators is therefore vital to the further research development of technical communication's history.

Rear Admiral Grace Brewster Murray Hopper would make an excellent addition to the canon of celebrated technical communicators. A Vassar-educated mathematician turned professor turned Naval officer, Grace was a formidable intellect and one of the earliest American computing geniuses; and as a technical writer, her work should be more widely examined.

The corpus of Grace Hopper's technical writing is worthy of study for several reasons: first, because little of her personal correspondence survives, her strengths as a writer must instead be directly evaluated on the basis of her work (Grudin and Williams 17). Second, Hopper herself was first a mathematician and an educator—her instructional habits and writing skills were well-developed before she was ever employed on the ASCC project (for example, Mitchell 1994). Third, Hopper worked at an exciting time in history, at the dawn of the computing age, using programming concepts that are still thriving today (Beyer 10, 16). Fourth, while much scholarship exists on Hopper's notable roles as a teacher, a programmer, and a Naval officer, as far as I am aware, no other research to date has explored Hopper's role or results as a technical communicator (Mitchell; Williams; Beyer *Information Age*). Finally, the *Manual of Operation for the Harvard Mark I ASCC* is a complete, substantive primary source which deserves deeper study. For all these reasons, Hopper's technical writing presents a unique opportunity rich with research value, as well as both inspiration and practical lessons for today's technical communicators.

Findings: Research Questions Answered

What makes the Mark I ASCC a worthy object of study? What is the least we need to know about the device to understand its manual and vice versa? In order to be useful to a modern reader, the components and processes of the Mark I ASCC required some introduction: punched cards, for example, are no longer used in computing. The ASCC was also unique among the early major mainframes, built using proven concepts and components from electromechanical engineering and telephony rather than with expensive vacuum tubes (for example, Williams 57, Beyer 32). Obsolete almost before it was fully built, the ASCC was uniquely designed and also the only computer of its era that solved equations via software like a modern computer (Beyer 52). Thus, illustrating how the ASCC solved problems in general, and how it solved differential equations specifically, was helpful in bridging the historical and conceptual knowledge gap between early mainframe technology and current computers. Last, but certainly not least, working knowledge of the Mark I ASCC is vital to understanding the purpose, shortcomings, and merits of Hopper's manual for it..

What makes the manual for the Mark I ASCC worthy of study as a technical document? The Mark I ASCC was the first American digital computer (Beyer 48). The *Manual of Operation* functions not just as an explanatory document, but also as a historical record of the machine itself (for example, Markel 13). Chapter III, Electrical Circuits, is of particular use to modern technical communicators decoding extinct hardware entirely from its documentation—it not only details the major physical components of the machine, but also explains how the computer solved problems, in per-cycle detail, literally down to the second (Hopper et al. *Mark I Manual* 10; 80 - 95). The manual's detailed diagrams and photographs of the computer's main components—relays and counters—help modern technical communicators understand these mechanisms. Relays and counters are shown disassembled, paired with labeled diagrams so that the interconnections of individual parts and their role in the machine can be better understood (Hopper et al. *Mark I Manual* XVI, XVII). Hopper's discussion of the main crankshaft mechanism, while densely detailed, illustrates physical action of the computer, which helps explain in practical terms some of the hardline physical limitations on the Mark I's processing speed (Hopper *Mark I Manual* 59, 60). Finally, the document has additional historical value as a substantive example of Hopper's early work and serves to place her later work in context.

What can Grace Hopper's approach to this manual, considered alongside a selection of her other writings, teach modern technical communicators? Grace's thorough-going approach to the *Manual of Operation*—written under crushing deadlines and wartime pressures at the bruising rate of five pages per day—offers inspiration and hope to student writers. Beyond her practical, take-no-prisoners workmanship, Hopper exhibited several additional strengths as a writer on the ASCC project. From the beginning, Grace ensured that the audience had sufficient historical and practical context for novel concepts, made certain to deliver multiple examples of each equation and coding type throughout, and always addressed instructions in terms of the audience's task necessities and future goals.

Hopper's work on the Mark I ASCC manual also leaves inspiration and instruction for future technical communicators in that the manual exhibited several features with room for improvement. Conciseness is the largest of these, and the favored modern approach: brevity is the manual's weakest feature. To improve conciseness, Hopper could have benefitted enormously from greater use of lists, use of positive constructions, and the elimination of extra prepositional phrases (Markel 14, 223, 237, 238; Carroll et al. 127, 129; Ganier 15, 18.) More thorough use of headings would have further offloaded the need for full-text description, as would additional manipulation of "mixed" diagram and document text (Markel 206; Ganier 18). Research has shown that these approaches reduce cognitive load for new users and decrease their "starting up" time when learning a new technology, software, or other device—and also would have distilled Hopper's writing while reducing its tonnage (Carroll et al. 128).

Hopper's collaborative articles with Aiken for *Electrical Engineering* did more than simply summarize the Mark I ASCC manual: these articles addressed the practical concerns and concepts necessary to hardware operation, as well as general principles for planning and executing software independently. The articles also left procedural writing guidelines for programmers looking to utilize the Mark I ASCC themselves. By providing compact, single examples of program logic and illustrating specific mathematical solutions in detail, Hopper concretized the subject in ways of interest to engineers and operators, stressing utility, conceptual mastery, and practical solutions (Aiken and Hopper *ASSC I* 389, 390; Aiken and Hopper *ASCC III* 325).

"The Education of a Computer", Hopper's later work on UNIVAC architecture, is something of a minor classic in software circles and also the first user documentation ever for software compilers, the foundation of the modern computational paradigm (Beyer 10, 223). This article showcases Hopper's clear evolution as a technical writer: here, we see extensive use of lists and "mixed" text and diagrams to communicate a wealth of conceptual information quickly over the span of two pages, the addition of brief orienting information before diving into complex tasks and concepts, and lists and headings organized to support the material (Markel; Carrol et al.; Ganier). By this point in her career, Hopper had clearly adopted streamlined principles of technical documentation that are still recommended as preferred strategies today (Markel; Carrol et al.; Ganier; Karreman and Steehouder).

Moreover, with "The Education of a Computer" Grace's mastery of context and core concept management is in full effect: she addresses mathematicians directly from the basis of her experience and understanding of their needs as computer users. Knowing their needs as mathematicians and the potential benefits for them of using the new technology, she begins with their mutual foundational exigence: how to solve harder problems in less time and with greater ease (for example, Hopper "Education" 244). From there, she addresses the minimum requirements a mathematician needs to manually solve a problem and then introduces novel factors, such as UNIVAC itself, subroutines, and machine addressing in order of complexity, and highlighting multiple uses for these concepts wherever possible (for example, Hopper "Education" 344, 345, 349). She does so in part to stress the versatility of compiling routines and the power of UNIVAC, as well as both the need for and benefits of automatic computing as an aim in the field (for example, Hopper "Education" 344, 347).

Opportunities for Future Research

In terms of addressing Grace Hopper's strengths and weaknesses as a technical communicator, much initial research has been done, but more remains. For one thing, the dense style of the *Manual of Operation* ensures that intensive examination of specific writing examples within could fill a dissertation (for example, Hopper et al. *Mark I Manual* 95, 98, 145, 181).

One of the largest areas for expansion of the current research is a more detailed and direct comparison of Hopper's collaborative articles with Aiken to the *Manual of Operation* itself: a specific one-to-one comparison of certain sections may highlight areas where a practical, condensed emphasis assisted with instruction completion, but obscured actual technical facts about the machine, and vice versa. Moreover, in the final article of the *Electrical Engineering* series, Hopper gives specific advice to mathematicians preparing their own programs and writing tape and plugging instructions for "operators" to follow (for example, Aiken and Hopper *ASCC 3* 235). There simply was not sufficient length to cover this aspect in more detail, though doing so would have enriched this thesis and indeed might have strengthened its central claims.

Future research might also position Grace Hopper's work alongside that of more modern technical communicators, especially women authors of note in the user documentation field. Consider Carol Kaehler, author of the original Macintosh user manual in 1984. Kaehler's common sense approach and accessible tone contributed to and enhanced Apple's reputation and consumer presentation. Kaehler described the graphical user interface (GUI) in familiar terms at a time when the notion of home computing was brand new, and explained actions as natural to modern users as double-clicking a mouse to a largely nontechnical novice audience who had never used—and perhaps never even seen—a computer mouse before. While sentences such as "You can start applications and get documents, work on them, and put them away again—just by moving the mouse and pressing the mouse button" are friendlier and more concise than "It is good practice to reset a storage counter just before using it. This frequently avoids the necessity of [using] starting tapes and preserves quantities in the machine as long as possible," Kaehler and Hopper have each done their level best to highlight useful information and explain the unknown in terms of its practical value (Kaehler 13; Hopper et al. *Manual of Operation* 109). Comparison of their work might thus draw a fuller portrait of women's influence in software user documentation, while also charting the progression of user-friendliness and task-orientation across two different, but equally dynamic, frontiers of the computer age.

Another fruitful route for potential future research should examine the rhetorical and ethical significance of Grace's work more directly: consider, for example, that the Mark I was expressly designed to complete equations that would ensure victory in war (for example, Beyer 30). Also, Aiken quite deliberately associated himself with Babbage's descendants in a bid for increased ethos by association, itself an angle worthy of more detailed examination—particularly in light of Aiken's bitter and longstanding feud with IBM director Thomas J. Watson. There simply was not scope here to consider neither the persuasive tenor of the manual, nor the interpersonal and ideological conflicts that may have shaped the larger projects of which the manual was a part.

It is also worth noting that Hopper has written several more articles than are examined in this thesis, as well as numerous speeches and conference papers on subjects near and dear to technical communicators, affording rich research opportunities for examination in further detail. While Hopper's work on the *Manual of Operation* has much to teach modern technical communicators, it is but one substantive example of her work, and one from early in her technical communication career. Further study of her later work may thus provide technical communicators with stronger insights and lessons, as a sort of master class in preparing instructions the way that "Amazing Grace" would. Indeed, instructors interested in designing such a class might focus on Hopper's thorough, deft approach to establishing sufficient context for new problems. Those interested in modeling the weaknesses of Hopper's writing as pitfalls to be avoided could easily point to conciseness and accessibility as areas with room for improvement. By incorporating other milestones from Hopper's work—including more than there were room or scope to examine in this thesis—interested instructors would have no shortage of examples from which to begin outlining either a tailored course plan or an avenue of further research.

In conclusion, Grace Hopper's experience and qualifications alone do not make her an excellent technical writer. Instead, it is her clear command of grammar and form, her understanding of her core audience's needs, and her tireless effort to drill in multiple examples and highlight several uses for each bit of information she teaches that shines forth from a study of her work. With a forthright approach, scrupulously exact grammar, and a powerful store of direct knowledge to draw from, Rear Admiral Grace Brewster Murray Hopper is a technical communication powerhouse—one who should be more widely read and better appreciated.

REFERENCES

Aiken, Howard H., and Grace M. Hopper. “The Automatic Sequence Controlled Calculator--I.” *Electrical Engineering*, (1946): 65 (8-9) 384-391, DOI: 10.1109/EE.1946.6434251, Accessed 18 Feb. 2014. Web.

--. “The Automatic Sequence Controlled Calculator--II.” *Electrical Engineering*, Volume 65, Number 10, pp. 449 – 454, DOI: 10.1109/EE.1946.6439869, Accessed 18 Feb. 2014. Web.

--. “The Automatic Sequence Controlled Calculator--III.” *Electrical Engineering*, (1946): 65 (11) pp. 522 – 528, DOI: 10.1109/EE.1946.6439921, Accessed 18 Feb. 2014. Web.

Allen, Jo. “Gender Issues in Technical Communication Studies: An Overview of the Implications for the Profession, Research, and Pedagogy.” *Journal of Business and Technical Communication*, 1991, vol. 5, no. 4, pp. 371-392, http://journals.sagepub.com.ezproxy.net.ucf.edu/doi/pdf/ 10.1177/1050651991005004003. Accessed 18 Feb 2014. Web.

Beyer, Kurt W. *Grace Hopper and the Invention of the Information Age.* Cambridge: The MIT Press, 2009. Print.

Campbell-Kelly, Martin, William Aspray, Nathan Ensmenger, and Jeffrey R. Yost. *Computer: A History of the Information Machine.* Boulder: Westview Press, 2013. Print.

Carroll, John M., et al. The Minimal Manual. Human-Computer Interaction, 1987 – 1988, vol. 3, no. 2, pp. 123 –153, http://eds.a.ebscohost.com.ezproxy.net.ucf.edu /eds/pdfviewer /pdfviewer? sid=60d152bb-a47c-42cb-923a-d543b6ef9248%40sessionmgr 4008 &vid=5&hid=4108, Accessed 21 Feb 2014. Web.

Ceruzzi, Paul Edward. The Prehistory of the Digital Computer, 1935 to 1945: A Cross-Cultural Study. Diss. U of Kansas, 1981. *ProQuest Dissertations and Theses Full-Text*, http: //ebscohost .com.ezproxy.net.ucf.edu/eds/detail/ detail?vid=2&sid=6a601de8-2fc5436e80ad9c75b00a021d %40sessionmgr4007&hid=4110&bdata= JnNpdGU9ZWRzLWxpdmUmc2Nvc GU9c2l0ZQ%3d%3d#AN=XISI17647-H&db=htm, Accessed 09 Dec. 2013. Web.

--. A History of Modern Computing. Cambridge: The MIT Press, 2002. Print.

Cohen, Bernard I. *Howard Aiken: Portrait of a Computer Pioneer*. Cambridge: The MIT Press, 2001. Print.

Einsohn, Amy. *The Copyeditor's Handbook: A Guide for Book Publishing and Corporate Communications, Third Ed.* Berkeley: University of California Press, 2011. Print.

Ganier, Franck. "Factors Affecting the Processing of Procedural Instructions: Implications for Document Design." *IEEE Transactions on Professional Communication*, vol. 47, no. 1, 2004, pp. 15-26, DOI: 10.1109/TPC.2004.824289, Accessed 18 Feb. 2014.

Grudin, Jonathan, and Gayna Williams. "Two Women Who Pioneered User-Centered Design." *Interactions*, vol. 20, no. 6, 2013, pp. 15 – 20, http://delivery.acm.org.ezproxy.net.ucf.edu/ 10.1145/2540000/2530538/p15grudin.pdf?ip=132.170.219.53&id=2530538, Accessed 18 Feb 2014.

Hackos, Jo Ann T. *Information Development: Managing Your Documentation Projects, Portfolio, and People*. Indianapolis: Wiley, 2007. Print.

Hopper, Grace Murray, and John W. Mauchly. "Influence of Programming Techniques on the Design of Computers." *Proceedings of the I.E.E.E., 1984.* 72 (9): 1214 – 1218. Print.

Hopper, Grace Murray, and the Staff of the Computation Laboratory. *A Manual of Operation for the Automatic Sequence Controlled Calculator.* Cambridge: Harvard University Press, 1946. Print.

Hopper, Grace Murray. “The Education of a Computer.” *Proceedings of the ACM National Meeting, 1952.* Pittsburgh: ACM, 1952. 243-249. Print.

Hopper, Grace Murray. “Possible Futures and Present Actions.” *Proceedings of the Australian Computer Conference, 1972.* Brisbane: ACM, 1972. 272 – 273. Print.

Johnson-Eilola, Johndan and Stuart A. Selber (Eds.). *Central Works in Technical Communication.* New York: Oxford University Press, 2004. Print.

Kaehler, Carol. *Macintosh*. Cupertino: Apple Computers, Inc., 1984. Print.

Karreman, Joyce and Michaël Steehouder. “Some Effects of System Information in Instructions for Use.” *IEEE Transactions on Professional Communication*, vol. 47, no. 1, 2004, pp. 34 – 43, DOI: 10.1109/TPC.2004.824286, Accessed 13 Sep. 2013.

Kemp, Juliet. “Grace Hopper and UNIVAC: Before There Was COBOL.” *LinuxVoice* May 2014: 82 - 85.

Malone, Edward A. "Historical Studies of Technical Communication in the United States and England: A Fifteen-Year Retrospection and Guide to Resources." *IEEE Transactions on Professional Communication*, vol. 50 no. 4, 2007, pp. 331 – 351, http://login.ezproxy.net.ucf.edu/login?auth=shibb&url=http://search.ebscohost.com.ezproxy.net.ucf.edu/login.aspx?direct=true&db=edsggr&AN=edsgcl.175157068&site=eds-live&scope=site, Accessed 18 Jan 2014.

Markel, Mike. *Technical Communication.* 10th ed. Boston: Bedford/St. Martin’s Press, 2012. Print.

Mitchell, Carmen Lois. (1994). The Contributions of Grace Hopper to Computer Science and Computer Education. Diss. U of North Texas, 1994. *ProQuest Dissertations and Theses FullText,* https://login.ezproxy.net.ucf.edu/login?url=http://search.proquest.com.ezproxy.net.ucf.edu/docview/304143024?accountid=10003. Web. Accessed 19 Dec. 2013.

Ramey, John Wilson. "The Coffee Planter of Saint Domingo: A Technical Manual for the Caribbean Slave Owner." *Technial Communication Quarterly*, vol. 23, no. 2, pp. 141-159, DOI: 10.1080/10572252.2013.811164, Accessed Apr. 10 2015.

Rubens, Philip. *Science and Technical Writing: A Manual of Style.* 2nd ed. New York: Routledge, 2001. Print.

Rauch, Susan. "The Accreditation of Hildegard von Bingen as Medieval Technical Writer." *Journal of Technical Writing and Communication*, vol. 42, no. 4, 2012, pp.393 – 411, http://ebscohost.com. ezproxy.net.ucf .edu/eds/detail/detail?sid=a6861c17-781a-4f30-87f5-7e6610e20c2c, Accessed 5 May 2014.

Sammet, Jean. "The Early History of COBOL." *Communications of the ACM: ACM SIGPLAN Notices Special Issue— History of Programming* Languages, vol. 13 no. 8, 1978, pp. 121–161, DOI: 10.1145, Accessed 15 Jan. 2014.

Shirk, Henrietta Nickels. "Contributions to Botany, the Female Science, by Two Eighteenth-Century Women Technical Communicators." *Technical Communication Quarterly*, vol 6, no. 3, 1997, pp. 293 – 312, DOI: 10.1207/s15427625tcq0603_5, Accessed 19 Dec. 2013.

Tebeaux, Elizabeth and Moran, Michael G. "A Bibliography of Works Published in the History of Professional Communication from 1994-2009: Part 1." *Journal of Technical Writing* *and Communication*, vol. 41, no. 2, 2011, pp. 193–214, DOI: 10.2190/TW.41.2.f, Accessed 19 Dec. 2013.

Tebeaux, Elizabeth and Mary M. Lay. “Images of Women in Technical Books from the English Renaissance.” *IEEE Transactions on Professional Communication*, vol. 35, no. 4, 1992, pp. 196-207, DOI: 10.1109/47.180280, Accessed 19 Dec. 2013.

Todd, Jeff. “Teaching the History of Technical Communication: A Lesson with Franklin and Hoover.” *Journal of Technical Writing and Communication,*vol. 33, no.1, 2003, pp. 65 – 81, http://eds.a.ebscohost.com.ezproxy.net.ucf.edu/eds/pdfviewer/pdfviewer?sid=d0a7506d-d604-4733-b9a3-1e79bd2b952b%40sessionmgr4009&vid=2&hid=4205, Accessed 8 Jan 2015.

United States National Defense Research Committee. *A Manual of Calibration Measurements of Sonar Equipment.* Washington, D. C. 1944. Print.

Voss, Christopher A. “Learning from the First Operations Management Textbook.” *Journal of Operations Management*, vol. 25, no. 2, 2007, pp. 239 – 247, http://dx.doi.org.ezproxy.net.ucf.edu/10.1016 /j.jom.2006.05.013, Accessed 8 Jan 2015.

Williams, Kathleen Broome. *Grace Hopper: Admiral of the Cyber Sea.* Annapolis: Naval Institute Press, 2004. Print.