THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 6, July–August 1983
Printed in U.S.A.

Human Factors and Behavioral Science:

The *UNIX*™ Writer's Workbench Software: Rationale And Design

By N. H. MACDONALD*

(Manuscript received December 17, 1981)

The *UNIX*™ Writer's Workbench software is a set of computer programs that help with two stages of document production: evaluation and editing. These programs analyze prose documents and suggest improvements. There are several types of programs: those that proofread, analyze style, and reformat the text in new ways, and those that provide information about the English language. This paper first describes the rhetorical and psychological writing principles that underlie the Writer's Workbench programs. It then describes the major Writer's Workbench programs and how they judge writing, based on these writing principles. Finally, it presents the human factors principles used in the design and development of the Writer's Workbench system.

I. INTRODUCTION

The previous paper in this issue of the *Journal*[1] pointed out the growing need for automated language processing, that is, for tools to help authors write clearly and understandably. Such tools are especially important for technical writing where the content is precise and a reader's failure to understand the text can be costly.[2] This paper

* Bell Laboratories.

describes how the *UNIX** Writer's Workbench software attempts to meet this need.

The first major section of this paper describes the principles of good writing that are incorporated into the Writer's Workbench programs. Many of these principles are supported by research showing that, indeed, different writing styles make a difference to the reader. The next section describes the major Writer's Workbench programs and relates them to the principles. The last major section describes the user considerations that guided the design of the Writer's Workbench programs.

II. SOME PRINCIPLES OF GOOD WRITING STYLE

This section presents some principles of good writing style. Research suggests that text that violates these principles is more difficult to comprehend. These principles belong to the word, sentence, paragraph, or document level, but of course, some principles overlap categories.

2.1 Word level

There are several word level principles besides the obvious ones such as using and spelling words correctly.

2.1.1 Wordiness

Style guides usually advise writers to avoid hackneyed, empty, or frequently misused phrases, such as "at this point in time" and "notwithstanding the fact that."

2.1.2 Definite, specific, concrete language

Strunk and White[3] remark that "the surest way to arouse and hold the attention of the reader is by being specific, definite, and concrete." They illustrate this principle by this pair of sentences, the first vague, the second concrete:

> A period of unfavorable weather set in.
> It rained every day for a week.

Psychological research on memory and the imageability (ability to create an image) of words suggests that texts with many abstract words will be more difficult to remember, and presumably to understand, than texts with concrete words.[4–8]

2.1.3 Word frequency and length

Coleman reported a high negative correlation between the average

---

* Trademark of Bell Laboratories.

frequency of content words in a passage and its difficulty to readers.[9] That is, texts made up of many infrequent words are difficult to comprehend. He also found that difficult passages contained more letters, syllables, and morphemes per word. Since frequency and length are highly correlated,[10] these two effects are possibly one effect. In keeping with these findings, Klare recommends Anglo-Saxon-based words over Latin-based words in English since the Latin form is usually longer, as in "go" versus "proceed."[11]

2.2 Sentence level

There are several obvious sentence-level principles, for instance, using correct punctuation and correct grammar. Some of the less obvious or more subjective principles are described below. These are not principles of right or wrong, but rather of better or worse.

2.2.1 Passive voice

One of the biggest writing problems, particularly in scientific writing, is the overuse of the passive voice. Historically the passive voice was used to indicate the objectivity of science and the scientist. The scientist did not state, "I found that...," but rather, "It was found that. . . ." This usage was dogma; as Einstein said, "When a man is talking about scientific subjects, the little word 'I' should play no part in his exposition."[12]

One problem with the passive voice in scientific materials is that its use has spread from obscuring "I" and "we" to many other cases as well, e.g.,

> A variable-gain control is included in this circuit.

Scientific objectivity is still served by stating:

> This circuit includes a variable-gain control.

Perhaps because of a change in scientists' perceptions of their role,[13] or perhaps because of the difficulty of the passive style, many scientists and scientific editors[14,15] now recognize and even promote the use of the active voice, including the use of first-person pronouns.

Why should we avoid the passive voice? The rhetoric books describe it as "dron[ing] like nothing under the sun," wordy and unclear,[16] and less direct and less vigorous[3] than the active voice. It may indeed be all those things, but in addition, psychological research has shown that the active versions of a sentence are recalled better[17] and verified faster.[18] Scientific texts written in the third person passive, as "It was concluded that . . ." are remembered less well and appreciated less than the same content written in the active voice.[19]

Kirkman[20] took samples of scientific papers and rewrote the content in six different styles. He asked scientists and engineers to rate which version they found "most comfortable to read, easiest to grasp and simplest to digest." In three different surveys, he found they preferred "direct, active writing, with a minimum of specialist vocabulary, a judicious mixture of personal and impersonal constructions, short and uncomplicated sentences and liberal paragraphing."

These data do not imply that passive voice should never be used; at times it is preferable to

1. Emphasize the object of the sentence
2. Vary the rhythm of the text
3. Avoid naming an unimportant actor.

EXAMPLE: The mail was delivered.

However, the passive voice should be restricted to the useful and necessary cases, rather than used widely and indiscriminately.

2.2.2 Nominalizations

Nominalizations are nouns that have been created from verbs. They usually end in "ion," "ment," "ence," or "ance," e.g., "transformation," "establishment," and "admittance." The empirical case against using nominalizations is strong. Coleman[21] found that individual sentences without nominalizations were remembered better than their nominalized forms. A multiple-choice comprehension test on content failed to show a significant difference between the two versions. But on a memory task, subjects required significantly more exposures to memorize sentences containing two nominalizations than to memorize the same content written in active-verb form.

In a later experiment, Coleman[17] investigated ten different types of nominalizations and their active versions. He found that for those pairs of sentences in which the active version was phrased in two clauses and the nominalized version in one, the active was memorized more quickly. For example

ACTIVE: If he discusses the reason for the price-change, it will be appreciated.

NOMINALIZED: His discussion of the reason for the price-change will be appreciated.

Coleman[21] took passages from a psychology text and rewrote all the nominalizations, passive sentences, and adjectivalizations (verbs formed from adjectives) into active sentences. (Coleman[17] found no difference between adjectivalizations and their active forms.) He found

that students answered correctly 25 percent more questions from the rewritten texts than from the originals. In a similar experiment, subjects were asked to write the passages immediately after reading them. Subjects recalled the simplified passages significantly better than the originals.

2.2.3 Expletives

In grammar, the term expletive refers to a syllable, word, or phrase that adds no information. In particular "expletives" are words such as "it" or "there," which anticipate a later word or phrase. Thus, in "There are three solutions to this puzzle," "There" is an expletive anticipating "solutions."

Many times such expletives can be deleted, e.g., "This puzzle has three solutions," although sometimes they cannot, e.g., "It is raining." Although no research demonstrates that expletives make text more difficult, Brogan[22] argues that when the expletive deemphasizes the main verb inappropriately, the sentence should be changed. For instance,

> It is this necessity that adds to their complexity.

can be changed to

> This necessity adds to their complexity.

making "adds" more salient.

2.3 Paragraph level

2.3.1 Readability

The readability or reading grade score for a text predicts how many years of schooling a reader would need to understand it. (Units other than years of schooling are sometimes used.) The prediction is usually based on the length of the words in the text and the length of the sentences. Different readability formulas calculate the lengths differently and weight the factors differently.

As we mentioned in the previous section, the length of a word (highly related to its frequency) predicts its difficulty. Sentence length is related to sentence type, with complicated sentences usually containing more words than simple ones. Readability formulas predict reasonably well the difficulty of the text, not because sentence length and word length cause reading difficulty, but because they are highly correlated with features such as complexity and frequency, which do.

As with any predictor, these formulas can be fooled. The Dale-Chall formula[23] takes vocabulary items into consideration, but most formulas

do not and will provide incorrect readability scores for nonsense text. All the formulas will give the same values for text with the sentences input backwards or forwards. But, in general, the formulas give a reasonable prediction of text difficulty when presented with naturally written text.

Unfortunately, research has shown[11] that the comprehensibility of the text is not necessarily improved, although the reading grade score is, by simply shortening words and sentences. The best way to improve the comprehensibility of a text is to rewrite it following the principles of good writing.

2.3.2 Variation

In writing, as in most fields of endeavor, moderation is best. The previously described principles are not absolutes; some passives and nominalizations are reasonable, and in fact, variation in sentence length, structure, and type is necessary to make writing interesting and keep the reader's attention.[13] There are other more important reasons to vary sentence type, which usually varies length as well.

Writing instructors suggest that less important ideas should be grammatically subordinated to more important ones so that the grammatical structure emphasizes the logical structure. Two simple sentences can be joined by using a "that" clause or an adverb, such as "although," to subordinate one to the other. The less important sentence should be in the subordinate clause after the "that" clause or adverb. For example, the following sentences:

1. The short, simple sentence is the most comprehensible form for an individual sentence.
2. Overusing such sentences may make a document seem disjointed.

can be combined:

> Although the short, simple sentence is the most comprehensible form for an individual sentence, overusing such sentences may make a document seem disjointed.

The combined sentence subordinates sentence (1) to sentence (2), thus emphasizing that the information in sentence (2) is more important than that in sentence (1).

2.4 Document level

2.4.1 Organization

Most books on writing recommend that the first sentence of each paragraph present the topic of the paragraph or else provide a transition from the previous paragraph into a new topic.[3,13] If, for most paragraphs, the first sentence reflects the topic, then these sentences give the reader a reliable signpost to the meaning. Headings also provide signposts to topics and topic changes. A paper with good

headings and topic sentences can be skimmed easily and quickly, and even the person who reads every word will find it easier to follow.

2.4.2 Audience considerations

Style books strongly advise writers to know their audience and to write for them. This is particularly important for materials such as instruction manuals, which the reader needs to understand. Writing for the reader extends from questions of vocabulary and sentence structure to content and organization. For content and organization, Flower[24] gives particularly thorough advice.

III. PROGRAMS

Until recently, students of writing could use only books and teachers to help them. This has slowly been changing with the advent of computer programs to do some of the work. Most programs, however, have focused solely on checking and correcting spelling.[25] Several readability indices have also been automated, but in general, wire services, magazines, newspapers, and businesses still do not analyze their text with the computer, although it is often stored and edited in a computer.

Although not yet in widespread use, several text analysis systems do exist. Besides the Writer's Workbench programs[26,27] to be described here, other systems include EPISTLE,[28] an IBM project still in the research stage; JOURNALISM,[29] a University of Michigan system that provides feedback to journalism students; and CRES,[30] a Navy system to help improve the quality of technical manuals and training materials. The EPISTLE system is a business office system that will abstract the contents of incoming letters but will also correct grammatical errors in outgoing letters. JOURNALISM provides some proofreading, but because it is programmed with specific knowledge about the articles it evaluates, it is able to comment on the organization and content as well. CRES calculates the readability score for the text, flags uncommon and misspelled words and long sentences, and suggests simple replacements for difficult words and phrases.

The rest of this section describes some of the Writer's Workbench programs, focusing on those that most strongly reflect the principles discussed in Section II.

3.1 Proofreading: proofr

The most useful Writer's Workbench programs are in some ways the least interesting. Every writer can use proofreading help, since as we become more familiar with a piece of writing we become poorer at spotting errors in it.

The proofreading program, proofr, invokes five separate programs.

A three-line example of input text and its proofr output are shown in Fig. 1. Each of these five programs can be run individually, but are more conveniently run as a package. When run separately, some of the programs have capabilities not found when they are run as part of proofr. Each separate program will be discussed in turn.

3.1.1 Spelling: spellwwb

The spellwwb program, a spelling checker based on the *UNIX* system spell program,[31] allows users to have a personal dictionary of words.

```
INPUT:   Our report, "The Basic Fundamentals of Organizatonal Complexity",
         is enclosed.  Please send any recomended changes at your
         earliest convenience. thanks.

PROOFR   ********** SPELLING **********
OUTPUT:  Possible spelling errors in examplefile are:

         Organizatonal                recomended

         If any of these words are spelled correctly, later type
                   spelladd word1 word2 ... wordn
         to have them added to your spelldict file.

         ********** PUNCTUATION **********

         The punctuation in examplefile is first described.

         2 double quotes and 0 single quotes
         0 apostrophes
         0 left parentheses and 0 right ones

         The program next prints any sentence that it thinks is incorrectly
         punctuated and follows it by its correction.

         line 1
         OLD: Our report, "The Basic Fundamentals of Organizatonal Complexity",
         NEW: Our report, "The Basic Fundamentals of Organizatonal Complexity,"
         line 3
         OLD: earliest convenience. thanks.
         NEW: earliest convenience. Thanks.

         For more information about punctuation rules, type:

                        punctrules

         ********** DOUBLE WORDS **********

         For file examplefile:

         No double words found

         ********** WORD CHOICE **********

         Sentences with possibly wordy or misused phrases are listed next,
         followed by suggested revisions.

         beginning line 1 examplefile
         Our report, "The *[ Basic Fundamentals]* of Organizatonal Complexity",
         is enclosed.

         beginning line 2 examplefile
         Please send any recomended changes *[ at your earliest convenience]*.

         file examplefile: number of lines 3, number of phrases found 2

         ---------- Table of Substitutions ----------

         PHRASE                       SUBSTITUTION

         at your earliest convenience: use "soon" for " at your earliest convenience"
         basic fundamentals: use "fundamentals" for " basic fundamentals"

         ********** SPLIT INFINITIVES **********

         For file examplefile:

         No split infinitives found
```

Fig. 1—Input to and output from proofr program.

The program searches the input text and prints all words that are not in its dictionary or the user's dictionary.

In addition, the spellwwb program can be used interactively to correct spelling errors. The spellwwb program prompts the user with each possibly misspelled word. Responding to each, the user can:

1. Find all lines on which it appears in the file
2. Invoke another program to try to determine the correct spelling
3. Tell the program how to change the spelling
4. Leave it as it is
5. Save it in a personal dictionary of correct words.

The user can also specify that certain misspellings always be changed. For instance, a poor speller might store the correction "relevant" for "relevent." Then spellwwb makes all specified corrections and updates the user's personal spelling file.

3.1.2 Punctuation: punct

The punct program searches for simple punctuation errors. It recommends changes to:

1. Move commas and periods to the left of double quotes, and move semicolons and colons to the right of double quotes
2. Capitalize the first letter of sentences
3. Balance double and single quotes and parentheses.

The program enforces straightforward rules, not ones that require judgment, such as deciding whether a comma or semicolon is the appropriate mark. When punct finds an error, it prints the original line, followed by its correction of the line.

Some of the rules punct enforces are unfamiliar to many people, such as the relative position of double quotation marks with other marks of punctuation. Nevertheless, they are accepted standards of American English. The punct program directs a user who has made punctuation errors to use the program punctrules, which provides a list of pertinent punctuation rules. The user, thus, has easy access to reference information, which can be used to decide whether the suggested changes are appropriate.

3.1.3 Consecutive occurrences of the same word: double

Using context line editors for editing text increases the chance of having the same word twice in a row. The double program locates consecutive occurrences of the same word, which can be split across two lines.

3.1.4 Wordy phrasing: diction

The diction program, described by Cherry,[32] searches a text file for phrases that writing experts have classified as wordy or frequently

misused. The latest version of its dictionary also contains some phrases that may reflect a sexual bias. The program prints sentences containing such phrases and surrounds them with stars and brackets (*[ ]*). It then recommends substitutions for these phrases. For instance, for the phrase, "bring to a conclusion," it recommends using "conclude," "end," or "finish."

Users should have the flexibility of tailoring the word list to their environment. In general, "end," "stop," or "finish" may be good choices for "terminate," but for many people writing about software, "terminate" is the appropriate word. Such a user can direct the program to stop searching for "terminate" by creating a personal dictionary. Having a private dictionary of extra terms and suppressed terms increases the usefulness of the diction program.

3.1.5 Split infinitives: splitinf

The splitinf program uses a "parts of speech" analysis program, parts,[33] to find infinitives that are split by adverbs. Such split infinitives as in "to quickly decide" are the most common type. Since users may be unfamiliar with this error, those whose papers contain split infinitives are told to use splitrules, which will print grammatical information about split infinitives.

3.2 Stylistic analyses

The stylistic analysis programs provide information whose interpretation is less concrete than that given by the proofreading programs. Hence, the information is more difficult to use; following the advice faithfully can require a considerable amount of time.

3.2.1 Tabular stylistic information: style

The style program,[32] based on parts, provides 71 numbers describing the stylistic features of a text. The most important variables it reports are several readability indices (described more fully by Cherry[32]), information on the average length of the words and sentences, the distribution of sentence lengths, the grammatical types of sentences used, e.g., simple and complex, the percentage of verbs in the passive voice, the percentage of nouns that are nominalizations, and the number of sentences that begin with expletives.

The style output for this article, shown in Fig. 2, is more useful for research on the style of documents, however, than for helping inexperienced writers improve their writing style. The style table is difficult for many writers to interpret because

1. Users may not know the meaning of some terms, e.g., "expletive" and "nominalization."
2. Users frequently do not know whether the numerical values

```
readability grades:
        (Kincaid) 11.3  (auto) 12.6  (Coleman-Liau) 13.1  (Flesch) 13.2 (48.8)
sentence info:
        no. sent 240 no. wds 4636
        av sent leng 19.3 av word leng 5.18
        no. questions 1 no. imperatives 0
        no. content wds 2734  59.0%  av leng 6.72
        short sent (<14) 24% (58) long sent (>29)  9% (22)
        longest sent 64 wds at sent 150; shortest sent 4 wds at sent 70
sentence types:
        simple  42% (101) complex  38% (92)
        compound  7% (16) compound-complex  13% (31)
word usage:
        verb types as % of total verbs
        tobe  32% (170) aux  16% (85) inf  17% (89)
        passives as % of non-inf verbs  14% (63)
        types as % of total
        prep 10.5% (487) conj 3.8% (177) adv 4.2% (197)
        noun 28.0% (1296) adj 17.2% (797) pron 4.7% (220)
        nominalizations  2 % (90)
sentence beginnings:
        subject opener: noun (48) pron (28) pos (1) adj (35) art (57) tot  70%
        prep  13% (32) adv  6% (15)
        verb  1% (3)  sub_conj  6% (14) conj  2% (5)
        expletives  1% (2)
```

Fig. 2—Style program output for this paper.

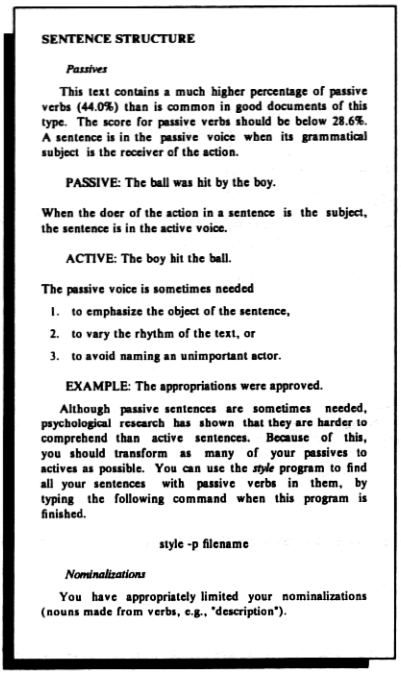
should be high or low, even for terms that are probably familiar, such as "complex sentence."

3. Users who know enough to minimize or maximize the use of some construction still do not know what numerical value is appropriate.

3.2.2 Interpreted stylistic analysis: prose

The prose program goes beyond the style program by providing the style statistics and an interpretation as well. The prose program compares the style values of the user's text against a set of standards and describes the differences in a two-to-three page output written in English sentences. A section of prose output is shown in Fig. 3.

Several sets of standards for comparison are available since texts are written for different types of readers and for different purposes. Users select which set of standards should be used in the interpretation of their text. Currently there are built-in standards for technical papers and prose training materials. These standards were set as follows. Department heads in the Bell Laboratories basic research area were asked to identify the best technical writers in their departments. These people were, in turn, asked to identify their best written technical documents (content aside). This process yielded twenty-eight documents. An editor in a Bell Laboratories training department provided 34 documents that he judged to be particularly well written. The technical and training documents were then run through the style program. The means and standard deviations of each of the style



SENTENCE STRUCTURE

Passives

This text contains a much higher percentage of passive verbs (44.0%) than is common in good documents of this type. The score for passive verbs should be below 28.6%. A sentence is in the passive voice when its grammatical subject is the receiver of the action.

PASSIVE: The ball was hit by the boy.

When the doer of the action in a sentence is the subject, the sentence is in the active voice.

ACTIVE: The boy hit the ball.

The passive voice is sometimes needed

1. to emphasize the object of the sentence,
2. to vary the rhythm of the text, or
3. to avoid naming an unimportant actor.

EXAMPLE: The appropriations were approved.

Although passive sentences are sometimes needed, psychological research has shown that they are harder to comprehend than active sentences. Because of this, you should transform as many of your passives to actives as possible. You can use the *style* program to find all your sentences with passive verbs in them, by typing the following command when this program is finished.

style -p filename

Nominalizations

You have appropriately limited your nominalizations (nouns made from verbs, e.g., "description").

Fig. 3—Section of prose program output for a poor paper.

variables were computed for each document set. These means and standard deviations now make up the standard for that document type. When the value of a user's variable is more than one standard deviation from the mean of that standard, prose recommends changes. The mkstand program can be used to create additional standards from a set of documents. This flexibility allows any writing group to tailor the standards for its particular audience.

If any of the input text's values is more than one standard deviation from the selected standard, prose explains why this may make the document hard to comprehend and how to rewrite the text to remove the problem. If scores on a variable are acceptable, and the variable is an important one, prose tells the user that the text achieved an appropriate score.

3.2.3 Stylistic problems in context: findbe

Although the style program can isolate individual sentences that contain passive verbs, expletives, or nominalizations, it is frequently difficult to know which sentences to change and how to change them without seeing the surrounding context. The findbe program automates part of a prescription for revision given by Lanham, who tells the

revisor to circle all forms of the verb "to be" and to try to replace them.[34,35] This advice agrees with some of the information presented earlier since "to be" occurs with passives, e.g.,

The difficulty of the passive voice *IS disregarded* by many writers.

with many nominalizations, e.g.,

Coleman's *discussion* of nominalizations *IS* comprehensive.

and with expletives, e.g.,

There ARE many ways to avoid expletives.

The findbe program underlines and capitalizes all forms of "to be." The text is then formatted as usual, providing the user with a paper that is easy to read, since all sentences are in context and problem areas are highlighted. This turns out to be a useful way of looking at the first draft of a paper.

3.2.4 Checking text abstractness: abst

The abst program[36] indexes the conceptual abstractness of the input text. Abstractness is defined here as the percentage of words in the text that also occur on a list of 314 words rated as abstract in psychological research.

When the percentage of abstract words is over 2.3 percent, the program suggests that concrete examples be introduced to make the document more understandable. (This cutoff was derived from the collection of good documents used to develop the prose standards.) The abstract words found in the text are saved in a file for the user to review.

3.3 Organization

3.3.1 Judging organization: org

The organization of a text is important since an appropriate structure makes it more comprehensible. For a computer program to analyze the organization of a text fully, it would need to abstract the content. Without a parser for English, or some other way of interpreting the meaning of a text, our programs cannot give feedback on the quality of the content and organization.

The org program, however, was designed to give a writer a different perspective on a text as an aid in evaluating its organization. The program formats the text and preserves headings and paragraph

boundaries, but prints only the first and last sentence of each paragraph.

For writers who follow the traditional topic sentence and concluding sentence format for paragraphs, the output can be a good abstract of the paper. Even for writers with a more casual style, seeing the overall structure of a long paper can help to improve it.

IV. USER CONSIDERATIONS

In discussing human-computer interactions, two issues should be considered: the ease of using the programs and the quality of the computer's responses.[37] This section will describe how the Writer's Workbench system attempts to optimize both aspects of the interaction.

4.1 User interactions

4.1.1 Program power

Perhaps the most important organizational decision made was to design the Writer's Workbench system hierarchically. One command, wwb, runs the most commonly used programs, the proofreading program, proofr, and the English-language stylistic program, prose. The wwb program is easily remembered as the acronym for the Writer's Workbench system, and so the casual user of the system finds the system simple to access.

For the experienced user of the Writer's Workbench programs, proofr and prose can be used individually, as can the separate components of proofr, thus allowing such users all the power they need. For one program, proofr, there is an alternate spelling, proofer, because based on the pronunciation it was often misspelled this way.

Many of the programs allow users to have their own dictionaries to tailor the output. Rather than requiring the users to type the names of the dictionaries as part of the command line, these programs use a particular file if it exists. Of course, users can still override this when they choose. Furthermore, commands were created to add words to these dictionaries, rather than requiring the users to edit them and keep them in sorted order. Thus, casual users can create personal spelling dictionaries and access them automatically with just two commands.

4.1.2 Documentation

For new users, there is now a substantial amount of paper documentation that describes the programs, how to use them, and their relation to good writing. But because users frequently do not own the paper documentation or do not have it near them, there are many on-line aids as well.

As with all *UNIX* system commands there are manual pages for each Writer's Workbench program, which are helpful to experienced *UNIX* system users. For the casual user, there is an on-line introductory system, which describes the uses of each Writer's Workbench program command.

Two other commands give on-line help. The wwbinfo command provides a list of all the commands and their functions. The wwbhelp command takes a key word as an argument and lists all the programs that have anything to do with that key word. These are words such as the following: "syllable," "prose," "passive," "sentence," and "organization."

Each Writer's Workbench program can also be run with a "flags" option, which prints that program's format and options. Further, since many of the programs have default options, these options are also echoed back with the user-specified options, as described in Section 4.2.2.

Finally, the output of many of the programs suggests other programs that would be useful. These suggestions are based on the analyses of the input text, and can thus remind users of programs that are useful for that text. For instance, users with spelling errors are told how to add correctly spelled words to their personal spelling dictionary, and the prose output suggests other appropriate programs. For example, users with too many passives are told how to find all sentences with passive verbs by using the style program.

4.2 Computer responses

What users have to remember and what they have to type are certainly important variables to consider when evaluating a system. But the quality of the messages the computer provides is also important.

4.2.1 Output length

The Writer's Workbench programs have attracted many new *UNIX* system users. For such users, the traditional "silence" of *UNIX* system commands is unfriendly,[38] e.g., if the spell command finds no misspelled words it simply stops, and the user receives a prompt (not a pat on the back). This silence is exactly what the regular *UNIX* system user wants, as McIlroy states, "Canned chitchat, like the plastic announcements on airplanes, may please newcomers, but it annoys old hands."[39] The obvious problem is that regular and casual users share machines, and what is right for one is frustrating for the other.

The first versions of most of the Writer's Workbench programs were verbose, irritatingly so, for experienced *UNIX* system users. We added "-s" options (for short) to most programs, which removed the

chitchat and most of the "pats on the back." Recently, we changed all the programs so that users can specify once what length of output they want in the future. Since expert users are best able to change this default, users who do not specify any length receive the long version.

4.2.2 Feedback

The Writer's Workbench programs inform the user when the program has started. Even experienced users may want reassurance that everything is proceeding on a heavily loaded system. Most commands echo the command line and include any options that were not chosen by the user but came about through default. This gives users a record of the exact command run; it may also remind them that they do not want a particular default, and it may alert them to options they did not know existed.

The Writer's Workbench programs cannot correct incorrect entries, but they try to provide complete, informative, and accurate error messages.

4.3 Needed improvements

To date, proofr is a major proofreading package, geared toward users who print the output on paper and make any recommended changes themselves. Since the output refers to a text problem by labeling it with the line number in the unformatted file, a user who has someone else type the text and only has the formatted output does not find these line numbers useful. For users with CRT display terminals, the output can be too lengthy to fit in the terminal's memory so that by the end of the program the first part of the output has disappeared. This makes it difficult to notice all the problems and change them.

We are currently designing two new versions of the proofr program. A version for word processing center customers will provide all the proofreading comments superimposed on the formatted output. For CRT users, we plan a completely interactive version, which will move linearly through the file and display possible corrections for the user to accept or ignore.

V. SUMMARY

This paper described some principles of good writing and some experimental results that show that readers prefer writing that embodies these principles and find it easier to understand. The paper then described a set of computer programs called the *UNIX* Writer's Workbench software. These programs help a writer isolate problems with general style as well as with individual sentences, phrases, and words. The final section of the paper described some of the human

factors principles that guided the design of the programs. The following paper in this journal[40] will describe the reception of the Writer's Workbench programs at Bell Laboratories and at two trial locations.

REFERENCES

1. L. T. Frase, "The *UNIX*™ Writer's Workbench Software: Philosophy," B.S.T.J., this issue.
2. J. P. Kincaid, R. P. Fishburne, R. L. Rogers, and B. S. Chissom, "Derivation of New Readability Formulas (Automated Readability Index, Fog Count, and Flesch Reading Ease Formula) for Navy Enlisted Personnel," Navy Training Command Research Branch Report 8-75 (February 1975).
3. W. Strunk, Jr. and E. B. White, *The Elements of Style*, New York: The Macmillan Co., 1959.
4. A. Paivio, "A Factor-Analytic Study of Word Attributes and Verbal Learning," J. Verbal Learning and Verbal Behavior, *7,* No. 1 (February 1968), pp. 41–9.
5. G. Frincke, "Word Characteristics, Associative-Relatedness, and the Free-Recall of Nouns," J. Verb. Learn. Verb. Behav., *7,* No. 2 (April 1968), pp. 366–72.
6. W. A. Winnick and K. Kressel, "Tachistoscopic Recognition Thresholds, Paired-Associate Learning, and Immediate Recall as a Function of Abstractness-Concreteness and Word Frequency," J. Exp. Psychol., *70,* No. 2 (August 1965), pp. 163–8.
7. A. M. Gorman, "Recognition Memory for Nouns as a Function of Abstractness and Frequency," J. Exp. Psychol., *61,* No. 1 (January 1961), pp. 23–9.
8. R. C. Anderson and R. W. Kulhavy, "Imagery and Prose Learning," J. Ed. Psychol., *63,* No. 3 (June 1972), pp. 242–3.
9. E. B. Coleman, "Developing a Technology of Written Instruction: Some Determiners of the Complexity of Prose," In E. Z. Rothkopf and P. E. Johnson (Eds.), *Verbal Learning Research and the Technology of Written Instruction*, New York: Teachers College Press, 1971, pp. 155–204.
10. G. K. Zipf, *Human Behavior and the Principle of Least Effort*, Cambridge, MA: Addison-Wesley, 1949.
11. G. R. Klare, *A Manual for Readable Writing*, Glen Burnie, MD: Rem Company, 1953.
12. A. Einstein, *Essays in Science*, New York: The Philosophical Library, 1949, cited in Mills and Walter, p. 29.
13. G. H. Mills and J. A. Walter, *Technical Writing*, 4th ed., New York: Holt, Rinehart, and Winston, 1962.
14. R. A. Day, *How to Write and Publish a Scientific Paper*, Philadelphia: ISI Press, 1979, p. 119.
15. M. O'Connor, *The Scientist as Editor: Guidelines for Editors of Books and Journals*, New York: John Wiley and Sons, Inc., 1979, p. 163.
16. S. Baker, *The Practical Stylist*, New York: Thomas Y. Crowell Co., 1962.
17. E. B. Coleman, "Learning of Prose Written in Four Grammatical Transformations," J. Appl. Psych., *49,* No. 5 (October 1965), pp. 332–41.
18. P. B. Gough, "Grammatical Transformations and Speed of Understanding," J. Verb. Learn. Verb. Behav., *4,* No. 2 (April 1965), pp. 107–11.
19. R. D. Ramsey, "Grammatical Voice and Person in Technical Writing: Results of a Survey," J. Tech. Writing Comm., *10,* No. 2 (1980), pp. 109–13.
20. J. Kirkman, *Good Style for Scientific and Engineering Writing*, London: Pitman, 1980.
21. E. B. Coleman, "The Comprehensibility of Several Grammatical Transformations," J. Appl. Psychol., *48,* No. 3 (June 1964), pp. 186–90.
22. J. A. Brogan, *Clear Technical Writing*, New York: McGraw-Hill, Inc., 1973, pp. 147–8.
23. E. Dale and J. S. Chall, "A Formula for Predicting Readability: Instructions," Educ. Res. Bull., *27,* No. 1 (January 1948), pp. 37–54.
24. L. Flower, *Problem-Solving Strategies for Writing*, New York: Harcourt Brace Jovanovich, Inc., 1981.
25. J. L. Peterson, "Computer Programs for Detecting and Correcting Spelling Errors," Commun. ACM, *23,* No. 12 (December 1980), pp. 676–87.
26. N. H. Macdonald, "Pattern Matching and Language Analyses as Editing Supports," The Amer. Educ. Res. Assn., Boston, April 1980.
27. N. H. Macdonald, L. T. Frase, P. S. Gingrich, and S. A. Keenan, "Writer's

Workbench: Computer Aids for Text Analysis," IEEE Trans. Commun., Special Issue on Communications in the Automated Office, *30,* No. 1 (January 1982), pp. 105–10.
28. G. E. Heidorn, K. Jensen, L. A. Miller, R. J. Byrd, and M. S. Chodorow, "The EPISTLE Text-Critiquing System," IBM System J., *21,* No. 3 (1982), pp. 305–26.
29. R. L. Bishop, "The JOURNALISM Programs: Help for the Weary Writer," Creative Computing, *1,* No. 2 (January/February 1975), pp. 28–30.
30. J. P. Kincaid, J. A. Aagard, J. W. O'Hara, and L. K. Cottrell, "Computer Readability Editing System," IEEE Trans. on Professional Commun. *PC-24,* No. 1 (March 1981), pp. 38–41.
31. M. D. McIlroy, "Development of a Spelling List," IEEE Trans. Commun., Special Issue on Communications in the Automated Office, *30,* No. 1 (January 1982), pp. 91–9.
32. L. L. Cherry, "Writing Tools," IEEE Trans. Commun., Special Issue on Communications in the Automated Office, *30,* No. 1 (January 1982), pp. 100–5.
33. L. L. Cherry, "PARTS—A System for Assigning Word Classes to English Text," Computing Science Technical Report, *81,* Bell Laboratories, Murray Hill, N. J. 07974, 1978.
34. R. A. Lanham, *Revising Prose,* New York: Charles Scribner's Sons, 1979.
35. R. A. Lanham, *Revising Business Prose,* New York: Charles Scribner's Sons, 1981.
36. L. T. Frase, P. S. Gingrich, and S. A. Keenan, "Computer Content Analysis and Writing Instruction," Amer. Educ. Res. Assn., Los Angeles, CA, April 17, 1981.
37. R. W. Bailey, *Human Performance Engineering: A Guide for System Designers,* New York: Prentice Hall, 1982.
38. D. A. Norman, "The Trouble with UNIX," Datamation, *27,* No. 12 (November 1981), pp. 139–50.
39. M. D. McIlroy, unpublished paper.
40. P. S. Gingrich, "The *UNIX*™ Writer's Workbench Software: Results of a Field Study," B.S.T.J., this issue.

AUTHOR

Nina H. Macdonald, A.B. (Linguistics), 1971, A.M. (Linguistics), 1974, and Ph.D. (Linguistics), 1979, University of Michigan; Bell Laboratories, 1976—. Ms. Macdonald joined Bell Laboratories first as a Resident Visitor in the Linguistics and Speech Analysis Department, studying the role of pauses in the perception of sentences. In 1979 she joined the Human Performance Engineering Department, where she has worked on the development of the Writer's Workbench programs. Member, Linguistics Society of America, Association for Computational Linguistics.