MANAGING COMPUTER CENTER DOCUMENTATION WITH AN INTEGRATED DATABASE

by

J. M. Caton
J. R. Sack
Stanford University
Stanford, California 94305

1. Introduction: The Problems of Computer Center Documentation

At Stanford, heavy emphasis is placed on interactive computing using locally developed software products. These products, most notably WYLBUR (text-editor), ORVYL (time-sharing monitor), and SPIRES (online database management), support instructional, research and administrative computing.

The "support" that any software product can give to a research, instructional or administrative project is largely dependent on the quality of documentation available for the product. What are the measures of "quality" in computer reference documentation? Here are a few suggestions:

1) Currency - Does the manual purchased today reflect today's system, or is it a year-old view of the system? Has the reprinting of an outdated manual been forced by lack of available time for revision?

2) Correctness - Are errors, small or large, going uncorrected because revision is a major task, and reprinting a minor one? Is the most correct information unavailable to users?

3) Completeness - Does a user need to read several different texts to get a complete users' view of the system? Are features documented in one text and not another? Is fragmentation a problem because it is easier to print an addendum or technical note than to incorporate new information properly into existing texts? Is there a lack of consistency among texts, perhaps because manuals for a single system are revised and republished at different times?

4) Efficiency for the reader - Can a user find needed tutorial or reference information without frustration? Is the manual readable? Can user suggestions for improvement be incorporated conveniently? Is a document formatted so that its appearance is a guide to the content rather than a mask?

5) Availability - When the Document Sales office closes, are users unable to access manuals? Are remote users served by remotely-obtainable documentation? Can a user access and print a revised document, perhaps after deciding that the current printed version is too out of date for use?

All of these barriers to quality documentation are real to writers of reference materials everywhere, but to computer documentation writers the continuous enhancement of software seems to require the continuous revision of software documentation.

2. Current State of the Art: The Computer Text Editor

The problems of timeliness, currency and availability were somewhat alleviated by the use of text-editor files. The old problems of typing and retyping were eliminated, but the other problems didn't go away. New problems arose, partially from an attempt to extend the text-editor into areas for which it was inappropriate. The text-editor was designed by programmers to supplant the card punch; it was not designed

by writers to supplant the typewriter. The text-editor's unit of content is the "line," which is an electronic "card" only longer. The writer's unit of content has always been the "sentence."

The transition from typewriters to text-editors brought several new burdens to the writer. Instead of one large "book" there are now many separate text-editor files, essentially several forms of documentation for one system, with each file somewhat duplicating materials in the others. There are reference and tutorial manuals, system overviews, command reference cards, online tutorial, and explain and help functions. The maintenance of consistency among all these separate files is a never ending task: as soon as one text is brought up to date, all the texts that refer to it are likely to become outdated.

The editing of the text of any one of these files is a chore in itself. If the text-editor is line- rather than sentence-oriented, several editing commands may be necessary to delete just a single sentence. After deleting the sentence, re-justification of the paragraph will be necessary; if justification isn't necessary, then the right margin may have to be realigned to prevent an unattractive sawtooth effect.

Of course, moving sentences or paragraphs requires that pagination be redone; this means taking out the old pagination and inserting the new. Other parts of a manual that refer to an altered part must also be checked for consistency, but how do you find all of those?

If several authors are working together, perhaps a writer and systems programmer, who trusts whom to alter or remove parts of a carefully integrated and formatted text-editor file?

Tonally, a text-editor file is completely neutral to its audience. The reader and author each see the same view of the text. Thus, the author must maintain any editing information, notes, gaps, bugs, dates and cross-references in a separate place, where it will not be seen by a reader.

All of these problems contribute to the great length of time between revisions of a document. Reprints are easiest; appendices and technical notes allow the overworked documentation staff to separate the up-to-date wheat from the out-of-date chaff. In a user community largely made up of novices, this lack of integration must be bewildering and frustrating, since it even confuses the documentation writer at times.

The problems of hand-typed documentation were legion; those of text-editor technology were fewer. There was a net functional gain, since virtually none of the typewriter's capabilities are missing from a complete text-editor. Any new approach to document management must guarantee that none of the existing text-editor capabilities should disappear.

3. SPIDOC: A New Approach to the Computer Management of Reference Documentation

SPIDOC (System for the Preparation and Inquiry of Documentation), a system developed and now coming into production use at Stanford, is an online document management system that relieves the writer of many of the mechanical and aesthetic burdens of text-editor technology. (see Caton-76a.) But SPIDOC is not simply a document formatting system; it is a document management system. It is not just a tool used by the author or authors of a document, but a facility for readers also. Early in its design phase, SPIDOC made a clear distinction between the needs of the author and the reader in a document management system. For readers, SPIDOC provides a "searchable" document that can be read hierarchically or subsetted online, and then perhaps published in part or in full, with automatic formatting, on a high-speed printer or at the user's terminal. For authors, it is a tool for preparing, maintaining and publishing text.

SPIDOC takes a database approach to structuring documents, using SPIRES (Stanford Public Information Retrieval System) as the database system. (See Schroeder-75.) Each "section" of a document is considered to be a "record" in a SPIRES database. The schema for the SPIDOC file system defines the way that sections will be linked to each other, and what information will be extracted from sections so that the information can be searched by readers. It is primarily these linkages and the unique data structures possible in SPIRES that provide the numerous facilities for authors and readers.

The author is required to break the document into logical sections, and, in actual practice, a section is often a "bite-size" part of the document for the reader--a page or less. By giving section headings to the logical parts of a document, and by structuring the sections so that

superior-subordinate (hierarchical) relationships exist among them, the author provides the reader with constant orientation by displaying the logical structure of the material. Thus, the criterion of "efficiency" of a document, its ease of use by a reader, is satisfied.

But several other SPIDOC features, particularly the types of automatic formatting, enhance the quality of a document by increasing its "efficiency." Although the author only enters the section number and heading with the section's body, where it is most easily managed, SPIDOC will generate a table of contents from these section numbers and headings, and link the individual sections to the page on which they begin. The page numbers are generated automatically when the document is extracted from the database ("published"), so repagination is never the author's problem. Justification and boldfacing is also done when the document is published. Other information is placed at the top and bottom of each page, allowing the reader to see the page, current section number and heading, and the date of publication.

As the author writes a section, he or she can determine that certain "key terms" or "indexing phrases" are described by material in this particular section. If these terms are noted, they are automatically accumulated into a sorted, multi-level index produced during publication.

Part of the design philosophy behind SPIDOC has been to eliminate duplication and fragmentation of information, and encourage multiple uses of one structured text file. This applies on all levels of the system. Information pertaining to a section is maintained only in the section itself, where it can be conveniently maintained by the author. It is the system's job to redistribute some of this localized information, to a table of contents or index for example, at publication time.

Other localized information is distributed by the system to searchable indexes or to other sections of the document. For example: if an author notes in one section that another section is referred to, the system generates an automatic cross-reference, and places a marker in the referenced section that indicates that it is referred to by a certain section. When the author goes to update information in one section, it is easy to see which other sections should be examined for possible revision.

The date of creation and date of revision are maintained and indexed automatically by the system. Not only does this allow an author to measure how far revisions (possibly by a co-author) have progressed, but it provides an invaluable aid to the reader who, perhaps having purchased a printed document some time ago, would like to retrieve and publish only sections of the document added or altered since that time. Of course, when only partial publication is done, pagination, the table of contents and the index refer only to the subset retrieved and published.

Other indexes are provided for the convenience of both reader and author. Indexes can be searched for key terms and phrases, section headings and type of material. This allows the reader or author to isolate only the subset of the manual that is relevant for his or her needs.

3.1 Classifying Text to Allow Precision Delivery of Reference Documentation to the Reader

SPIDOC allows the author to indicate the nature of the material in a section by specifying "section-types" or "group-types." (Groups are parts of sections, usually one or more paragraphs of similar intent.) Although the classifications are chosen by the author, at Stanford we have found several to be generally useful.

Section-types, which refer to an entire section and not just a few paragraphs, include: 1) Facility - for describing a potential or capability in the system; 2) Command - for describing the commands necessary to use a facility; 3) Option - for discussing command or facility options; 4) Message - for describing system messages that may be received when a particular command, facility, or option is used; 5) Figure - for tables and figures that must not be split over page boundaries; 6) Note - a "temporary" type, used by the author or authors to document bugs, mistakes, ideas, and any new material that is later to be incorporated into the other parts of a section.

Group-types include: 1) General - for top-level information, the description of a system's or feature's capabilities, and the kind of general information one might find in a glossary; 2) Formal - for statements of command syntax and options; 3) Detail - for explanations of command functions and options, suggestions for use, involved descriptions of capabilities; 4)

Example - for examples of a facility's use.

Dividing sections into groups with different types has not yet been exploited fully at Stanford. It appears that the extra structure this division gives complex material may be an aid to the reader's comprehension, and perhaps a good discipline for the writer. Another application of the concept of "group-type" allows masking of certain kinds of material from the reader based on the group's type. The author can freely insert notes and ideas into the text, using certain group-types for these memoranda and other insertions, and prevent a reader from seeing this information until the material has been properly incorporated into the text. Yet the new material is, for the authors' convenience, integrated into the document as a whole, since it is maintained within the section to which it applies.

By masking out all groups except "General," one should be able to publish a system overview. By masking out all groups except "Formal," one could publish a command reference card. Of course, some care must be taken in assigning and delimiting section and group-types before this kind of custom profile publication is feasible.

Multiple authorship is facilitated by providing each author with a group-type to flag his or her insertions. One author can search an index online and retrieve and examine sections that have been modified by another author. This material can be shared immediately by all authors.

Because of the SPIRES system "deferred queue," an author may make changes to a section and restore that section to its pre-modified form any time that day. Updates are not actually performed until the early morning hours. Few text-editors provide an author with a "second-thought" capability, unless authors use the confusing procedure of maintaining multiple versions in separate files.

3.2 Example of Specific Tools for the Author

Although the SPIDOC system makes a clear distinction between the reader and author, most of these distinctions are like two sides of a single coin. A single facility, such as group-type specification, provides different advantages to the reader and to the author.

For some facilities the advantages to the reader or author may seem less than apparent. One such facility, of primary advantage to the author, is "split sentence" formatting. Most writing is done in "block sentence" format, in which a new sentence appears on the same line as the end of the previous sentence. For example:

> This is the beginning of a paragraph in block format. Notice how sentences are strung together like segments of a tapeworm. Also, imagine how difficult it will be to change the order of segments in this worm using a line-ather than sentence-oriented editor. How many commands would it take to reverse the second and third sentences?

During the editing process, the SPIDOC author would usually choose to have these sentences displayed in "split" format. New sentences are always begun on new lines, and continuation lines of sentences are slightly indented, to allow the eye to scan material quickly. For example:

> This is the beginning of a
> paragraph in split format.
> Notice how sentences are
> isolated from each other in
> different lines.
> Also, image how easy it would
> be to change the order of
> the sentences in the
> paragraph, perhaps making
> this the second instead of
> the third sentence in the
> paragraph.

Notice how easy it is to spot a very long sentence. An editor could quickly make one long sentence into two shorter ones. Or, spotting several short sentences in a row, a writer may want to combine them for a smoother flow. The eye can easily determine when there is a healthy variation in sentence length, and a variety of sentence introductions and patterns. A writer's style is cast into a sort of "bas relief" during the editing process. Herein lies the advantage to the reader of split sentence formatting: better writing.

3.3 Online Access to the Documentation Database

In much of the discussion so far, we have referred to the reader as the potential publisher of a document. After isolating the sections of interest, or perhaps determining the entire document to be of interest, a reader issues a command that "publishes" the subset on the

high-speed printer. But this kind of access still does not serve the remote user well; a request for information must be serviced immediately, at the user's terminal.

SPIDOC provides a capability to serve a user's information requests at the terminal, by providing a hierarchically structured reading procedure. The user enters the reading program, and either indicates a particular section number at which reading is to begin, or, if no specific request is made, is shown a structured table of contents. From the table of contents, the user can quickly see where reading should begin. The table of contents orients the user to the document. It indicates sections up from the current reading location (3.1 is up from 3.1.2), sections at the same level as the current location (3.1.3 is at the same level as 3.1.2), and sections down from the current section (3.1.2.1 and 3.1.2.2 are down from 3.1.2). For example:

```
Up:      II How to Select and Search
            a Subfile
         II.1 Introduction
Prec:    II.2 Steps in the Search-
            ing Process

====>    II.3 Commands to Use in
            Selecting a Subfile
Down:       II.3.1 The SHOW SUB-
               FILES Command
            II.3.2 The EXPLAIN
               SUBFILE Com-
               mand
            II.3.3 The SELECT
               Command

Next:    II.4 The Basis of a
            Search Request
         II.5 Negotiating a Search
            Request
         II.6 Examining Records
            Found by Searching
```

Group-types can be specified in online information requests to make a more precise requirement for the kind of information needed, reducing the amount of irrelevant material that must be read. For example, an experienced user requiring aid may need to see only a formal command description, while a novice will need to see all information available. To serve the experienced user, a "REMIND" or "EXPLAIN" command could be implemented that would draw from the same information as the command reference and publication program. Again, the theme of consolidation: when information changes, only one file need be updated for the change to be made in all forms of documentation.

4. Review: Some Solutions to Computer Center Documentation Problems

Let's re-examine the attributes of "quality" computer documentation, this time with particular reference to facilities that a database approach such as SPIDOC offers over text-editor files. You will note that the dominant themes of the database approach are the facilitation of demand revision and demand publication. Demand revision means that an author, editor, or programmer, having decided that a part of a document is incomplete or inaccurate, goes into the document itself and leaves a note where all involved, authors and readers, can get to it. Demand publication means that an author or user who wishes to see part or all of a text may, at any time, issue only a few commands to get a document that is attractive and useful in format and content.

The measures of quality of database-maintained documentation are:

1) Currency - The user now has a choice: purchase a printed version, perhaps a few weeks old, at Document Sales, or invoke a publishing program that will produce a document incorporating the most recent revisions. Using simple search and retrieval commands, the user can determine what changes have been made to the document since its last printing. Printing may now occur more frequently, since changes are readily incorporated into the automatically reformatted text.

2) Correctness - Because revision is now a much simpler task, the practice of reprinting inaccurate documents because of lack of time for thorough revision will be less defensible. Correct information will become immediately available to users who wish to examine the document using online search and retrieval commands.

3) Completeness - The integration of new-feature documentation into existing text is readily accomplished. The stop-gap approach had been to provide "News" or "System Changes" files, or perhaps technical notes and appendices; this led to fragmentation of documentation sources, since new and old were kept apart. New information can now be properly incorporated into existing text.

A second kind of integration is facilitated by "group-type" specification. Top-level or glossary documentation, tutorial and reference documentation can all be placed together; certain types of documentation can then be masked out to publish a facility overview or a command reference card. This new approach, consolidation and integration, rather than isolation and fragmentation, makes the maintenance of all types of system documentation easier. Instead of all documentation of one kind being maintained together (tutorial documentation, for example), now all kinds of documentation of one component are maintained together in a single section of the text (facility overview, tutorial and command reference discussions of a "help" command, for example).

4) Efficiency for the reader - Important "format" guides to the content of a document, such as the table of contents, index, pagination and cross-references, are automatically generated when a document is published. Other helps to the reader's eye, such as boldfacing, are also provided automatically.

5) Availability - Via the online reading program, all users always have the most current version of a document available. An entire document or portions of it can be retrieved and then read or printed. When the Document Sales office closes, or when the current published version is superseded or out of stock, users still have access to a document. Stock copies of seldomly requested documents (some program library write-ups, perhaps) and back copies of old newsletters need not be kept. A user or document sales person can access the stored version and simply print the number of copies needed. No time and effort is required to keep an inventory, and no space is wasted on yellowing paper.

5. Conclusion: Toward a Cost-Effective Realization of the Documentation Database System

The Stanford computer facility has many remote users. Of the one hundred or so that are logged on at any one time during a weekday, only about half are on-campus. Some of the rest are in the city of Palo Alto, but many are in other cities and some are out-of-state. The need to provide the large and geographically diverse interactive community with timely and accurate computer documentation has led to the development of SPIDOC.

We currently have three major documents in the system, and plan to have all SPIRES user documentation (six manuals) in SPIDOC by the end of the year. The computer center's newsletter and the program library write-ups will probably be placed in the system soon after. A major off-campus user is considering the application of SPIDOC to its documentation problems.

The requirements for quality computer documentation have been addressed with some success by the SPIDOC system. In the near future, we hope to begin "tuning" those features that have shown the most promise for enduring, cost-effective use. We are aware of the pitfalls of measuring cost-effectiveness in an area where measurement is difficult (how much is good documentation worth?). But we realize that only in attempting to bring the system to functional "maturity" will we move closer to the goal of improving the timeliness, accessibility, and accuracy of computer-based text files.

BIBLIOGRAPHY

CATON-76a

J. M. Caton
The Integrated Computer Database as a Medium for Collecting and Referencing Textual Information
Ph.D. thesis, Department of Communication,
Stanford University, 1976

CATON-76b

J. M. Caton, J. R. Schroeder
An Integrated Database Approach to the Management and Use of Reference Documentation
Proceedings of Share XLVI,
Montreal, Quebec, August 1976 (in press).

SCHROEDER-75

J. Schroeder, W. Kiefer, R. Guertin, W. Berman
Stanford's Generalized Data Base System
International Conference on Very Large Data Bases,
Boston, Massachusetts, September 1975.