THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 61, No. 6, July–August 1982
Printed in U.S.A.

Automated Repair Service Bureau:

On-line Documentation: Mechanizing Development, Delivery, and Use

By R. J. GLUSHKO and M. H. BIANCHI

(Manuscript received June 17, 1981)

We describe the design and development of on-line documentation for a minicomputer-based management information system. We outline the design choices, compare on-line documents with paper ones, and review human engineering and "software psychology" issues. On-line documents are accessed from any dial-up terminal. Document retrieval shares a common user interface with other information activities like report generation, trouble reporting, and interuser communication. Documents are "modular" with properties that make them easier to create, use, and maintain.

I. INTRODUCTION

Surveys confirm that documentation does not always meet the information needs of computer system users. Documents often arrive later than the system they are supposed to support. Related information may be scattered through several documents, inadequately cross-referenced, or inconsistent. The users may need to rewrite documents to reflect local policy, tariffs, or company organization, adding more costs and delays to the expensive and time-consuming documentation process.

These problems with document timeliness and organization have grown increasingly severe in the last decade as computer systems have proliferated. The traditional procedures for developing and delivering paper documents have not kept pace with the increased information needed to support complex and volatile software products. Instead, with computer systems for text editing and storage becoming readily available, a growing number of Bell System applications are hoping to

resolve the mismatch between software systems and paper documentation by developing and delivering documents along with software in computer-based form.

Mechanizing the documentation process by using computer text editing systems and by treating documents the same as software seems like an obvious solution to the documentation problems for software systems (see Refs. 1 and 2.) Unfortunately, this solution can be as simplistic as it is obvious. Documentation problems rarely reduce to separate problems of development, delivery, or use. When we set out to create an on-line documentation system for the Cable Repair Administrative System (CRAS), we struggled to cope with the radical changes in the traditional ways of developing, delivering, and using documentation that came about when we transformed paper documents into computer-based ones. We found it challenging to create the tools and techniques we needed to coordinate the development of software and documentation. We emphasized eliminating the duplicated and disjointed efforts that carried over from methods for delivering software and documents in different media through separate channels. We worked hard to build a system that was easy to use for the telephone company personnel whose needs for information underlay our entire effort. We share some of our insights and hindsights in this paper.

The CRAS system generates management reports on the performance of telephone cable maintenance forces and the condition of the outside plant. The system runs under the *UNIX** operating system on a Digital Equipment Corporation VAX† 11/780 minicomputer. We will not discuss the role of CRAS in day-to-day telephone company operations (see Ref. 3)

II. DEFINITIONS AND GOALS FOR COMPUTER-BASED DOCUMENTATION SYSTEMS

Computer-based documentation is a broad goal that can be realized in many different ways. Different applications may face different documentation problems, user populations, or computer hardware that shape the design and goals for a mechanized documentation system. Before describing the specific system that was developed for CRAS, it will be helpful to define some of the varied forms that computer-based documentation may take to provide context for the specific design choices.

The documentation cycle begins with document definition, determining which documents need to be created. Next, these are written

* Trademark of Bell Laboratories.
† Trademark of Digital Equipment Corporation.

and edited. The finished documents are delivered to the users. Finally, the documents are used and maintained by the telephone companies, whose needs may lead to revised or new documents, new software or even to new systems. We distinguish three basic forms of computer-based documentation that differ primarily in how much of the entire documentation cycle they mechanize: the *electronic factory*, the *electronic library*, and *on-line documentation*.

The *electronic factory* is where computer-based documents are developed. Documents are captured at their source as they are written using computer text-editing systems, thus saving at least one costly and time-consuming translation from paper to computer-based form. The document development computer then serves as an electronic factory from which paper documents are manufactured as required. The same text source produces either typewritten or photocomposed output, and changes can be made with considerably less effort than for documents that exist only on paper. Note, however, that the intended user of the documents is not allowed direct access to the documentation in computer-based form.

The *electronic library* extends the electronic factory one step. It offers information on-line in a form organized for the intended users. The electronic library is typically a computer that serves as a central repository of documents, training materials, or planning information that users can access on-line from dial-up terminals. This is especially useful when the specific end users are unknown in advance and are dispersed in many different locations. However, the electronic library is logically, and usually physically, distinct from the computer system whose documentation it contains.

On-line documentation spans the usage stage of the documentation cycle by making information available to users in their normal work environment as they use a computer system at a terminal. The goal is to minimize the user's need to maintain paper documents, since the most reliable and current copies are always available on the computer. The *man* command that prints pages from the *UNIX* User's Manual is a rudimentary example of on-line documentation. To our knowledge, CRAS is the first telephone company operations support system with computer-based documentation that is on-line in development, delivery, and use.

III. COMPUTERS VERSUS PAPER

The potential for computer-based documentation systems to reduce the documentation problems of the telephone companies seems so obvious that it is almost impossible to imagine a Bell System application that would not benefit substantially from even the basic electronic

factory idea. Nevertheless, we resisted the temptation to "damn the paper, computerize at full speed" when we realized that transforming documents from paper to computer-based form had its costs, as well as its benefits. We will discuss some of the trade-offs that directed and constrained the CRAS approach.

Paper documents are "frozen" when they are written. Since they are not easily changed, they must be detailed and complete, hence the problem of bulky paper documents that contain more information than any one person ever seems to want. In contrast, the fundamental property of computer-based documents is that they can be easily changed. When coupled with selective display of information and flexible organization, computer-based documents can be thorough, yet easy to use and maintain.

On the other hand, it is important not to overlook the desirable properties of paper documents that are not possible to recreate on-line. Few computer terminals can reproduce pictures or graphics with the quality available in print. While low-cost graphics terminals someday may be commonplace, and computer systems far more universal and reliable, people will always be more familiar with paper documents. Only paper documents can be personalized with margin notes or "dog ears" to make them unique and useful information sources.

The complementary features of paper and computer-based documents suggest to us that even the most technologically advanced on-line documentation system will be surrounded by paper documents of its own creation. Every prophecy of a "paperless office" or "paperless society"[4,5] seems countered by more conservative forecasts in which paper remains in an important, albeit changed, role.[6,7,8] People will not quickly abandon all paper documents for electronic images that glare at them from television screens. Instead, users of on-line documentation systems will create some mix of computer and paper documents that takes advantage of the benefits of each medium. New users often bring with them inappropriate ideas of what computers can and cannot do, and these expectations may lead them to reluctant, inefficient, and, from a designer's standpoint, unpredictable uses of the computer. Many on-line documents will be printed and carried to the computer terminal in loose-leaf notebooks and dutifully replaced when new electronic versions appear.

We might respond to such uses of our on-line documentation system as weaknesses in its design to be overcome with some clever program or technique. Instead, we remind ourselves that we set out to meet the information needs of people, and note that technological issues seem to be secondary to the human engineering and "software psychology"[9] issues in the design and use of on-line documentation systems.[2,10]

IV. THE CRAS DOCUMENTATION SYSTEM

The general theme is that we seek to make documentation part of CRAS instead of just support for it. Using the computer to mechanize a system's documentation is a first step, but the challenge is to unify documentation and software, simplifying the user's interaction with the computer. We present four different perspectives on the CRAS documentation plan that define this objective:

- CRAS is an integrated information system. All of the CRAS information activities—report generation, document retrieval, trouble reporting, and interuser communication—share a common user interface.
- CRAS has a broadened view of documentation. Its on-line documentation contains the "standard" information that would be delivered with any operations system, but also contains information that might normally be called "performance aids" and "training." Locally-generated documents can be installed in the on-line system.
- CRAS documents tend to be "modular" with properties that make them more manageable and more easily maintained on the computer or in a user's notebook.
- CRAS documents are "computerized," as well as "computer-based"—the computer uses information implicit in the CRAS file structures and software to rewrite key documents and to provide a degree of "intelligence" to the document retrieval process.

4.1 The Cable Repair Administrative System as an integrated information system

The primary user of CRAS is an analyst who uses CRAS reports to understand and improve the condition of the telephone cables and the performance of cable repair forces. Let's observe a new analyst at the terminal using the various information subsystems of CRAS to see how the activities fit together and complement each other.

4.1.1 On-line performance aids

Our analyst wants to generate the CRAS Geographic Area Summary Report, Report 02. The analyst needs to be reminded by some documentation or performance aid how to specify the report request. The CRAS system provides both kinds of information when the analyst types *rpt02* at the terminal:

```
$ rpt02
Usage: rpt02 lev org period [-c geoid geoidvalue] [-t ncode4 number]

For more detail type: prtdoc cmd.rpt02
```

The "Usage" reminder is modeled after those provided by many

UNIX system commands. We expand its function by giving the user explicit instructions (the specific CRAS command) for retrieving the on-line document called *cmd.rpt02*. The "Usage" line is a "performance aid," a memory jog for experienced users. The other line shows less expert users the exact step for obtaining information about retrieving Report 02.

4.1.2 The "prtdoc" document retrieval command

The *prtdoc* command prints on-line documents at the analyst's computer terminal. The syntax of *prtdoc* is not the result of a "shotgun wedding" between document names and a retrieval procedure; we carefully organized and named the on-line documents to make them easy to retrieve. Overview and one-of-a-kind documents are retrieved by one-word names; for example, a table of contents that lists all on-line documents is retrieved by typing *prtdoc contents*. All other documents are arranged in categories with names made up of the category name and the document name, as in *cmd.rpt02*. All documents in a category are retrieved by typing just the category name (e.g., *prtdoc cmd*).

Prtdoc usually begins presenting the requested document to the user's terminal in five seconds or less, because documents are preformatted in files. Storing "finished" documents rather than source files takes up slightly more space, but saves the user the one-minute or more wait if the text formatting were done at the time of document retrieval. The storage space that CRAS documents occupy is a minuscule requirement on the computer sized for storing the large data bases needed to generate CRAS reports. Documents take up at most a few percent of the disk space in the standard CRAS configuration.

The *prtdoc* command has an option that prints any documents that have changed since a specified date. We provided this feature to help the many CRAS analysts who prefer to keep personalized paper copies of the most relevant documents in loose-leaf notebooks that they bring with them to the computer terminal.

Another useful option in *prtdoc* allows the user to find documents whose names, titles, or major section headings match key words. This helps users find information when they know what they are looking for but cannot remember the names of the relevant documents.

4.1.3 Other on-line information facilities: "crasprob" and "crasmail"

Suppose the analyst successfully generates the desired CRAS report, but suspects an error. A problem reporting command called *crasprob* allows the analyst to report problems or suggestions to the CRAS administrator and to support groups at Western Electric and Bell Laboratories. *Crasprob* maintains accurate records of all problem

reports, their status, and their solutions. *Crasprob* is efficient and natural because it is part of the analyst's work environment at the terminal, where problems are usually encountered.

Some time after the analyst submits the problem, the CRAS administrator or someone in the support organizations determines that the analyst's report is not in error, but contains some curious rules for counting data items. The CRAS administrator then invokes the *crasmail* command to notify all CRAS analysts throughout the company of the problem and its resolution. *Crasmail* also archives all interuser communication for future reference. When the analyst next types *mail* the answer to the problem will appear at the terminal.

If CRAS problems require changes to software and documentation, the CRAS support organization transmits the new information to CRAS installations over dial-up lines. The local CRAS administrator simultaneously installs the software and documentation with a single command, ensuring consistency. This coordination is possible because files containing documents and files containing software look identical to the computer.

Crasprob and *crasmail* are both built around the *UNIX* system's electronic *mail* and evolved from problem-reporting and interuser communication systems devised for CRAS development.

4.2 Broadened view of documentation

The CRAS system blurs the usual distinctions between information that might be called documentation and that normally classified as performance aids or training. The CRAS on-line documentation scheme contains information of all types that has the common purpose of enhancing the user's performance. We have already described the "standard" documents retrieved using the *prtdoc* command and the on-line usage reminders that tie software to documentation. Other kinds of information aids are as follows:

- *Report prompting*—Normally, when generating a report the analyst types the command name followed by any required or optional items. However, in report prompting the user types *prompt* followed by just the command name and the computer asks for each item in turn and continues until the analyst supplies enough information to generate the report correctly. Prompting takes more time, but the computer can determine whether the entries are valid as the analyst supplies them, so it is a valuable training aid for inexperienced users or a performance aid for entering complicated report commands.
- *Training data base*—The reference data base that is used to create sample reports for on-line documents is available for on-line training in report generation.
- *Local documents on-line*—Documents created by the telephone

company to reflect local procedures or policies, or "nonstandard" documentation, such as lists of names or telephone numbers, can be installed in the "standard" *prtdoc* on-line data base by the CRAS administrator.

- *Usage records and analysis*—All commands that analysts type at their computer terminals are recorded in a command log used by the CRAS administrator and support and development groups to improve CRAS performance. A statistical summary of the log is provided to the CRAS system administrator. This information about how CRAS is used has been the source of many of its most useful features. For example, we noticed that certain analysts periodically retrieved the table of contents of the on-line documents and then painstakingly studied it line-by-line to determine if any documents had been changed. We automated this task and made it an option in the *prtdoc* command.

4.3 Document modularity

The CRAS system contains about 250 on-line documents that average three pages in length. One popular definition of a software module is that it is a piece of code that "hides one secret." The analogous definition of a document module is that it is a unit of information that "tells one secret." For example, when a programmer creates a new piece of software, it is easy and natural to write a new document that describes it.

Not all CRAS documents are modular—nor should they be. One-of-a-kind documents like indexes and tables of contents are not broken up, and some overview documents that "glue" modules together are necessary. But in general, the modular organization of CRAS documents as short, self-contained units of information creates useful properties:

- *Terminal compatibility*—The CRAS documents are easy to use at a computer terminal. Even at the slowest display speeds, most documents are so short that they print in only a few minutes.
- *Work-unit organization*—Most documents contain the information needed for a single task or unit of work—retrieving a report, sending a message, or doing some other single activity. The CRAS changes over time as reports or commands are created, changed or deleted, but the modular organization of the documentation makes these changes easy to manage. Adding a command only requires adding a single short document, which is easily installed at the same time as the new software. There is no need to accumulate a large number of changes to justify reissuing a complete user manual.
- *Flexible job definition*—A job or position is made up of a collection of tasks or work units. For example, the analyst retrieves certain reports, analyzes them, and executes various communication and out-

put control commands at the terminal. The work-unit organization of CRAS documentation allows the analyst to have the information needed for the job, without any irrelevant information, even when companies define the analyst job differently. For example, Telephone Company A may make data base maintenance reports part of the analyst's job, while Telephone Company B may assign them to a clerk. Since each report is described in a separate document, analyst A and analyst B can have customized documentation manuals with just the right information, even though A and B perform different jobs.

- *Integrating local documentation*—Modular organization makes it easy to integrate local information with standard information. Local information that would not fit into a bulky paper manual naturally fits into collections of short pieces of information in an on-line system. In the past, telephone companies either maintained their local documents separate from standard documents or reissued standard documents at considerable cost.

4.4 Computerized, not just computer-based documentation

The CRAS system uses the computer for more than simply storing documents. Whenever possible, CRAS uses the computer to make documents more current, more reliable, and more useful. In particular, "dynamic" documents are generated by computer programs and use information implicit in the file structure or software to update themselves. Some of the most important of these are:

- *Table of contents of CRAS documents*—This is the master list of all documents that can be retrieved using the *prtdoc* command. The table lists them by the short name used to retrieve them, along with the complete title of the document, the number of pages it contains, and the date that it was last changed. The table is regenerated when any document is changed or added to the on-line documentation system. A related program automatically regenerates a permuted index to all the titles.
- *Dictionary of data fields*—This lists the elements of CRAS data base records that are used to generate the reports. If more types of information are added to the data base to allow for more specialized reports, this document is automatically updated from the data dictionary.
- *Run folder documents for host jobs*—These documents are the "Run Folders" or "Run Books" for several data base extraction jobs run for CRAS on a remote IBM computer. CRAS run folders are created from the Job Control Language for each job and are automatically customized to reflect the specific run-time options, sizes, and file names selected by the CRAS administrator.

V. OTHER LESSONS FROM CRAS ON-LINE DOCUMENTATION

5.1 Constraints on document development

Delivering documents along with software in CRAS required that programmers become part-time technical writers and increased our direct development costs. In most projects, documents are written by technical writers from other organizations, which may make them of higher quality, but which usually makes them late and less precise. The CRAS developers used the computer aids for writing and editing—the "Writer's Workbench"—that became available during the system's development.[11] We think that any trade-offs we made in document quality are far outweighed by gains in document timeliness and better coordination with software. Nevertheless, management should allocate greater resources to the development team and regard documentation as more than a nuisance activity beneath the efforts of programmers.

5.2 Easy compliance with document standards

On-line modular documentation easily lent itself to the new standards recently adopted by AT&T for Operations Systems Deliverable Documentation (OSDD). The OSDD recommends that documents be organized to meet the information needs of particular jobs or users rather than letting a single document contain "everything for everyone." Even though CRAS development was nearly complete when the OSDD standards were adopted, in a few weeks we devised customized collections of our document modules and created an on-line command that automatically printed the documents in OSDD form. In general, maintaining documents in computer-based form makes compliance with document standards much less painful than it has been in the past.

5.3 An adjunct dial-up advance information system

When documents are developed on-line in an "electronic factory," the intended users can have a preview that helps them plan while letting them provide valuable input to the development process. We provided a log-in on our development computer to telephone company personnel so that they could examine documents under development and experiment with the on-line retrieval system. They were better able to plan for CRAS, and by monitoring the dial-up use of the *prtdoc* command, we significantly improved the organization and usability of the documentation. This system became available almost without effort because we were developing the documents on-line anyway.

VI. PROSPECTS FOR ON-LINE DOCUMENTATION

The CRAS system is part of a larger network of many different kinds

of computer systems, terminals, and personnel subsystems. Some systems run on large computers like an IBM 370, others run on minicomputers in both stand-alone and networked configurations, and still others run on microprocessors. Dial-up asynchronous terminals used by analysts and managers sit next to dedicated synchronous terminals used by data entry clerks.

One form of computer-based documentation cannot work for every application—we know because we are facing the challenge of developing a plan for mechanizing the documentation of the entire family of systems of which CRAS is a part. Nevertheless, we have "deCRASsified" many of the programs and techniques used in CRAS, and they are being adopted by other projects as we work together toward that goal.

VII. ACKNOWLEDGMENTS

The CRAS documentation system was shaped by the collective efforts of the CRAS development group, who merged documents and software just as a motley collection of systems engineers, psychologists, and software developers merged and became the CRAS team in 1979–1980. In particular, J. R. Mashey, G. T. Vesonder, and K. A. Wright made substantial contributions to the work and ideas presented in this paper.

REFERENCES

1. K. L. Heninger, "Specifying software requirements for complex systems: New techniques and their application," IEEE Trans. Software Eng. (1980), *SE-6*, pp. 2–13.
2. A. I. Wasserman, "Information System Design Methodology," J. Amer. Soc. Inform. Sci., *31* (1980), pp. 5–24.
3. P. S. Boggs and J. R. Mashey, "Automated Repair Service Bureau: Cable Repair Administrative System," B.S.T.J., this issue.
4. E. Hanson, "Preparing our Thinking for the Year 2000," Word Processing World, *5* (1978), p. 40.
5. A. Toffler, *The Third Wave*, New York: William Morrow, 1980.
6. W. Benedon, "The Paperless Society: Fact or Fiction?" Inform. and Records Management, *13* (1979), pp. 101–2.
7. V. A. Vyssotsky, "Computer Systems: More Evolution Than Revolution," Journal of Systems Management, *31* (1980), pp. 21–7.
8. D. Wallace and P. Yates-Mercer, "A Paperless Society: Reality or Myth? Management Services, *24* (1980), pp. 12–7.
9. B. Schneiderman, *Software Psychology: Human Factors in Computer and Information Systems*, Cambridge, Mass.: Winthrop, 1980.
10. S. Hiltz and M. Turoff, *The Network Nation: Human Communication Via Computer*, Reading, Mass.: Addison-Wesley, 1978.
11. L. T. Frase, "Computer Aids for Writing and Text Design," Symposium presented at the Annual Meeting of the American Educational Research Association, Boston, Massachusetts, April, 1980.