

# Learning By Doing It On Your Own

By Hans van der Meij

What did you do when you had your first shot at a computer program like, say, a word processor?

- *Did you try to get the thing to work for you as you explored how it worked?*

or

- *Did you first study the 400 odd pages of the manual that accompanied the program?*

If you are in the latter group, you are a minority. Approximately three of every four adults want to get to the business of word processing as quickly as possible. Trying things out, or exploratory behavior as it is called, is not a very effective way to learn new things, however. In general, learning is much more effective if people develop a conceptual framework or model before they start practicing what they have learned.

Good manuals should try to support both of the above approaches to learning. On the one hand, they ought to accommodate the needs and skills of the learners. That is, they should support exploratory behavior, and focus on what the users can do with the program rather than on what the program can do. On the other hand, manuals should also support the development of a conceptual model. They should give users deep insights to make them independent problem solvers, enabling them to also discover new ways or uses for the program.

Until recently, technical writers were not very effective at solving this instructional problem and mock reactions like "when all else fails consult the manual" were not without ground. But times have changed, and more and more manuals have become a real help for first-time computer users. How this development was brought about and what these manuals may look like is the topic of this article.

## Times Are Changing

There are various reasons why technical writers have been enabled to write better manuals.

**Time and money.** Managers have found out the hard way that manual development is worth more than an afternoon of the programmers' time. Good manuals can sell programs, make customers happy, and relieve the help desk. Unfortunately, they also require special skills and considerable time to write.

**Theoretical models.** Theoretical models have been advanced that can assist the development of the manual. For example, the GOMS Model of Card,

(Moran & Newell, 1983) is a powerful scheme for classifying (part of) the behavior of first-time users. In addition, it gives writers insight into the problems that may arise when teaching users various methods to achieve a single goal as opposed to teaching only one such method.

**Appropriate content.** Third, it has become known what users *want from* and *do with* manuals. Experiments and observations have yielded fundamental insights about manual usage. These insights are summarized below.

**The Penguin Syndrome.** Many adults do not read manuals cover to cover as they are expected to (Sullivan & Flowers, 1986; Mack, Lewis, & Carroll, 1983). Instead, they tend to skip and skim them, studying only the information that they find relevant. Conventional manuals often cause the "Penguin Syndrome," coined after the following letter from an 11 year old boy to an editor:

*Dear Sir,  
I am returning this book because it told me more about penguins than I ever wanted to know.*

**Reading to do.** Many adults, especially those without an academic background, do not like to be placed into the role of a reader "conceptualizing the program." They want to try things out, get hands-on experience (Sticht, 1985; Redish, 1988; Carter & Honeywell, 1991). Conventional manuals tend to overemphasize the need for the development of a conceptual model and put the user into the undesired role of being a reader.

**Keep it simple, please.** First-time users do not study the manual in order to find out which procedure is faster. Efficiency is not their concern. Instead, they want to learn procedures that are easy to perform and remember and which can be applied in many situations. Usually, this means that users ask for a very limited set of methods. Conventional manuals introduce far too many methods. Research on Unix has, for example, indicated that of the 400 procedures that are available, users typically used only 20 regularly. The other 380 methods were either not used at all or employed infrequently.

**If anything can go wrong, it will.** Murphy's Law aptly describes what first-time users are likely to experience. Failures, mistakes, and crashes are part and parcel of what most first-time users will go through. Conventional manuals tend to assume that

users make no mistakes. The directions are supposedly perfectly clear, they have been made "idiot proof." All the readers must do is play them back.

**Active exploration.** Adults are not passive consumers. Placed before a computer, most adults easily slip into an exploratory mode, fiddling around with the keyboard to discover the program. Conventional manuals hardly ever accommodate this exploratory behavior. In fact, they try to keep it within bounds as much as possible, because it is an inefficient and sometimes even ineffective learning strategy.

In short, novices want mostly to get hands-on experiences with a program and are reluctant merely to read about it. In addition, they want to explore the possibilities of the program by themselves to see how it may benefit their work. Users are also likely to make many mistakes. Within certain limits, the manual must adapt to these user behaviors. That is, it should focus on the development of procedural skills rather than on declarative knowledge. Also, the manual should support exploration and prescribe what users must do as little as possible. And, finally, the manual must give users plenty of help (and time) for error recognition and recovery. The minimal manual tries to accommodate these behaviors.

### Characteristics Of The Minimal Manual

A minimal manual (MM) is based on four major principles. It is action-oriented, optimizes the use of text, supports error recognition and recovery, and it repeats little. These principles will be illustrated with the aid of a prototypical chapter from a MM that Lazonder and Van der Meij (1992) developed for a word processor (see Figure 1). A similar chapter from the Commercial Manual (CM)—made by the manufacturer—is presented as an example of a conventional manual (see Figure 2).

Both manuals are focused on getting the user to *read to do* rather than to *read to learn*. There are, however, clear differences in the way these manuals support user actions. The *action-oriented* nature of the MM becomes clear from the following features:

Users are prompted for action in the manual early on. The MM has short, if any, preambles such as a preface, introduction, summary, or organizer. Figure 1 presents the second chapter of the manual which is preceded by no more than four introductory pages. In contrast, the chapter from the CM is preceded by a "Getting Started" booklet which consists of a three-page preface, nine pages of text about keyboards, pull-down menus, menu options, codes, and a brief introductory lesson of four pages.

The MM mentions special terms (e.g., floppy, formatting, cursor, status line, word wrapping, and so on) only when they are necessary for task execution. Such terms are explained in the proximity of the task and their description focuses on what they "do." A nice example of how the MM treats jargon is shown with the word cursor which is explained

just before users need it. Conventional manuals tend to pull special terms out of context, highlighting them in separate sections. The "Getting Started" booklet of the CM in which users are asked to read about pull-down menus and the like is typical. Conventional manuals also frequently introduce jargon without good reason. A good example of this is the "word wrapping" mentioned in the CM. In view of the automatic nature of the carriage return, it is at the very least futile, but more probably obstructive, to mention such a term.

The MM supports the development of sound basic routines for elementary procedures. It offers users only that information which helps them develop a good working method for basic word processing routines. In the CM, the information is flawed. Users are taught to print a text before saving it. This is simply an unsafe routine.

The MM stimulates users to explore new possibilities of the program through sections called "Try on Your Own." These sections mention several additional options that are closely related to but not identical with the one that is practiced in the chapter.

In order to optimize the *use of text* to the actions of the users, there are a number of basic principles for the design of a MM.

There is little text. To reduce the text, a MM does not have advance organizers, summaries, or reviews. More importantly, the MM does not spell-out everything that happens. Readers are prompted to search for the relevant cues on the screen rather than having them spelled out in the manual. For example, after typing the note, the CM asks the reader to attend to the cursor, carriage return, possible differences with the example, and the status line. In contrast, the MM merely explains the cursor. The other concepts are irrelevant at that time.

There is no index either. The MM adapts to and activates the users' prior knowledge as much as possible. It does so in the headings of the manuals, for example. These headings are goal-oriented headings to refresh the users' memory and easy retrieval. Key-oriented headings such as "Delete," "Home," and "F10" found in conventional manuals are meaningless for the novice.

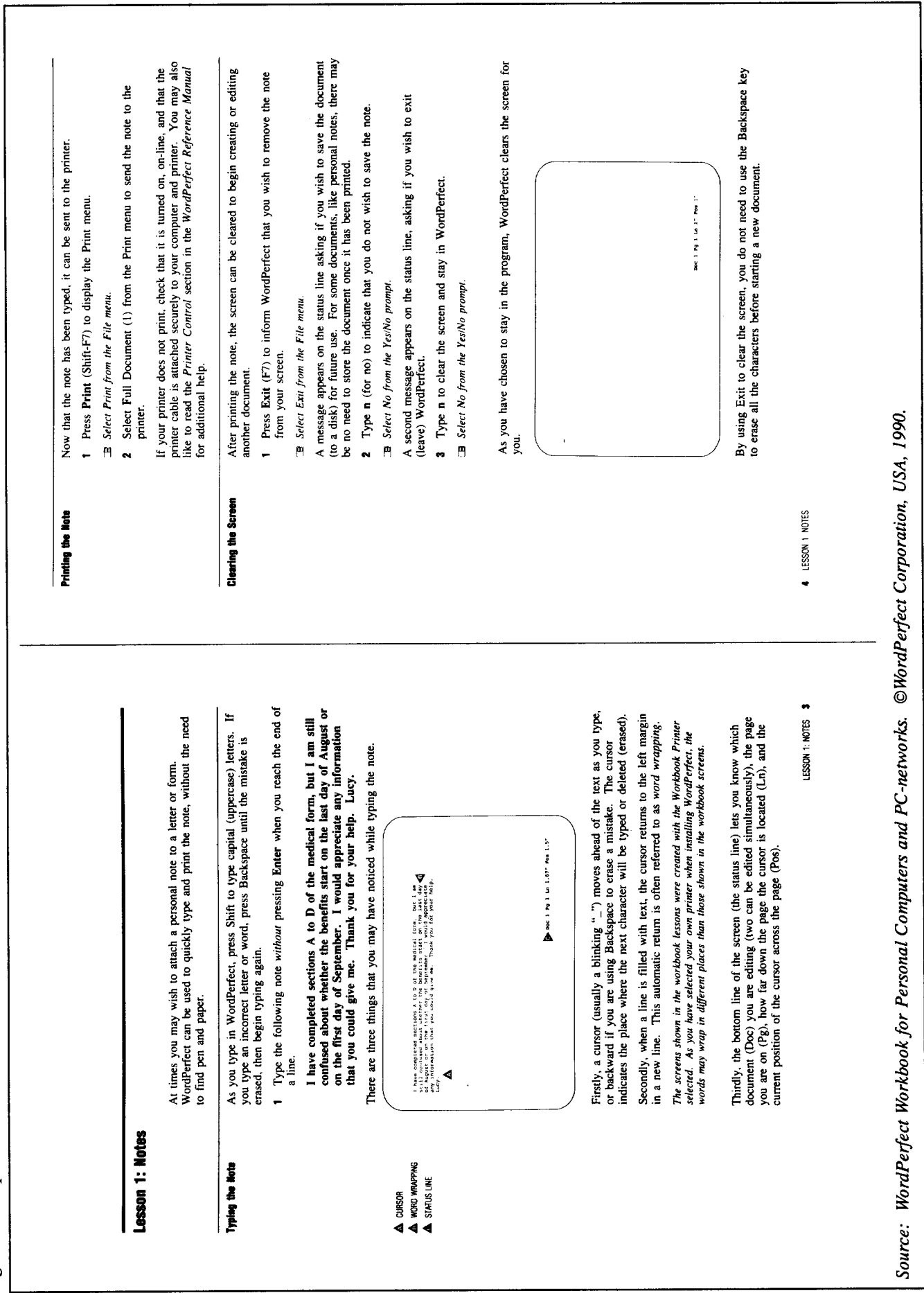
The MM has relatively many short sentences. The average sentence length is about 12 to 14 words, aiming at a reading level of 11 to 12 year olds (as most advertisements do). One could not find a sentence like "As you have selected . . ." (see Figure 2) in an MM.

Another key feature of the MM is the support for *error recognition* and *recovery* by the user. In view of the fact that most users make mistakes, such support is essential. The most striking principles for this support are the following:

- General error recovery mechanisms should be treated before specific ones. In the MM the users learn the most general recovery mechanisms in

Page No. 20 Missing

Figure 2. Chapter 1 Of A Conventional Manual



the first chapter. They learn how to start and end the program, and they are introduced to the two magic "Undo" buttons with which nearly all of the actions in the word processor can be undone. Like most conventional manuals, the CM seems to depart from the idea that users do not make any mistakes.

- The MM signals the error recognition and recovery information to ease user's location within the text. In the CM this information (i.e., correcting with Backspace) is hidden in the running text. As noted in Figure 1, the error recognition and recovery information is always placed immediately after the incorrect action, and is always in italics.
- The MM often prompts users to look at the screen and to search for the relevant information. These cues are meant to prevent users from making mistakes and may help error recovery since it is known that novices usually have difficulties with allocating their attention properly. In contrast, users of the CM are not directed toward the screen.

The most problematic feature of the MM is its *modularity*. We would prefer that each chapter stand alone so that it can be consulted without knowledge of the other chapters. However, we have found it impossible to follow this strategy. Instead, the modularity of the MM rests on two principles: closure and subtle referencing.

All but the first chapter have the same beginning and end. They start with retrieving a document and end with saving the revised file. One might object that this means a lot of needless repetition for the user. But our research shows it has a very positive effect on these managerial tasks which are the most difficult to learn.

Users are not advised to consult pages further on or earlier in the manual. Referencing is much more subtle: Users are merely informed of the necessary goals that must be accomplished before their present task. Thus, they are prompted to search the manual for the goal-directed (i.e., identical) heading.

## Conclusion

Do these new ideas really work? Do they help first-time users get acquainted with computer programs better than before? They do. The research literature even suggests incredible gains over conventional manuals (e.g., Black, Carroll, & McGuigan, 1987; Carroll, Smith-Kerker, Ford, & Mazur-Rimetz, 1988; Gong & Elkerton, 1990; Ramsay & Oatley, 1991). For example, learning time is often reduced by a third and users solve about 25 percent more problems correctly after training with an MM than with a CM. With an MM users thus become much more proficient in a shorter period of time.

The enormous gains of the MM in comparison with the CM suggests that we are on a new track

for the design of manuals for first-time users. Already the differences with "conventional manuals" have become more subtle. This is all the more reason to study how valid the design principles of the minimal manual really are and how far they can be generalized to other self-instructional materials. We are optimistic. The ideas advanced by the minimalists are, for example, quite compatible with the constructivists who call for a new approach to education (e.g., Duffy & Jonassen, 1991). Like minimalists, the constructivists emphasize the need to activate the learner's prior knowledge as much as possible, to urge them to follow their own plans and goals, to construct their own ideas, and to reflect about what they are doing. In short, to construct meaning from their experiences. This is exactly what users are urged to do with the aid of a minimal manual. ■

## References

- Black, J. B., Carroll, J. M., & McGuigan, S.M. (1987). What kind of minimal instruction manual is the most effective? In H. J. Bullinger, B. Shackel, & K. Kornwachs (Eds.), *Proceedings of the Second IFIP Conference on Human-Computer Interaction*, p. 159-162. Amsterdam, NH: Elsevier.
- Card, S. K., Moran, T. P., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, N.J.: Lawrence Erlbaum.
- Carroll, J. M., Smith-Kerker, P. L., Ford, J. R., & Mazur-Rimetz, S. A. (1988). The minimal manual. *Human-Computer Interaction*, 3, p. 123-153.
- Carter, J. H., & Honeywell, R. (1991, February). Training older adults to use computers. *Performance & Instruction*, p. 9-15.
- Duffy, T. M., & Jonassen, D. H. (1991). Constructivism: New implications for instructional technology? *Educational Technology*, 31, p. 7-12.
- Gong, R., & Elkerton, J. (1990). Designing minimal documentation using a GOMS model: A usability evaluation of an engineering approach. In J. Carrasco Chew, & J. Whiteside (Eds.), *Proceedings of the CHI '90 Conference*, p. 99-106. New York: ACM.
- Lazonder, A., & Van der Meij, H. (1992). *Towards an operational definition of the minimal manual*. IST-MEMO 9202, University of Twente.
- Mack, R. L., Lewis, C. H., & Carroll, J. M. (1983). Learning to use word processors: Problems and prospects. *ACM Transactions on Office Systems*, 1, p. 254-271.
- Ramsay, J. E., & Oatley, K. (1991). Designing minimal computer manuals from scratch. *Computer and Writing*, 4, p. 86-106.
- Redish, J. C. (1988). Reading to learn to do. *The Technical Writing Teacher*, 15, p. 223-233.
- Sticht, T. (1985). Understanding readers and their uses of texts. In T. M. Duffy, & R. H. Waller (Eds.), *Designing usable texts*. New York: Academic Press, p. 315-340.
- Sullivan, P., & Flower, L. (1986). How do users read computer manuals? Some protocol contributions to writers' knowledge. In B. T. Petersen (Ed.), *Convergences: Transactions in reading and writing*. Urbana, IL: National Council of Teachers of English, p. 163-178.

## About The Author

Hans van der Meij specializes in instructional design. In cooperation with Ard Lazonder, he has conducted a number of experiments on minimalist manuals. Their recent studies have focused on the design and effects of error-information in tutorials. Address: University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands.