

Ten Misconceptions about Minimalism

John M. Carroll, *Senior Member, IEEE*, and Hans van der Meij

Abstract—We describe ten common misconceptions about the minimalist approach to documentation design. For each, we analyze how the misconception arises from plausible interpretations of minimalist principles and heuristics. We then clarify how each misconception deviates from minimalism, as we understand it. Analysis and discussion of creative elaborations of minimalism—including “misconceptions”—can promote a sharper concept of what minimalism is.

THE minimalist approach to designing instruction and documentation relies on task orientation to produce more effective learning and performance outcomes more rapidly. It emerged through the course of the last decade as a broad attempt to bring theory, empirical research, and practical design experience to bear on design practice. Accordingly, it is at once a body of psychological theory, a body of empirical research, and a body of design practice and discourse. The approach has attracted attention from a diverse group of researchers and practitioners with a variety of interests and disciplinary backgrounds. Not surprisingly, this has created an expansive and sometimes confusing literature pertaining to minimalism.

In this paper, we try to provide focus for the continuing discussion on minimalism by critiquing aspects of the extant literature. To begin, we acknowledge that what we will call “misconceptions” are in every case understandable as legitimate interpretations of minimalism. Frequently, they are simplifications of what we intend by minimalism, useful in particular cases. We applaud the initiative and creativity of our colleagues in helping to explore and develop the concept of minimalism. Nevertheless, it is important also to occasionally rein in the concept, question the simplifications, call attention to inconsistencies and contradictions. Our hope is that this effort will cause more, not less debate, research, and practical development of minimalism.

The central principle in minimalism is task orientation. But many other principles play a role in this design approach either because they support task orientation or because they follow from it [1]. One could hardly expect otherwise of a broad design philosophy. However, this architecture can be problematic: there is a temptation to see minimalism as constituted entirely of one or another of its constituent principles.

1. Minimalism means brevity.
2. Minimalism means incomplete instructional analyses.
3. Minimalism means trial-and-error learning.
4. Minimalism does not support people who learn by reading.
5. Minimalism overemphasizes errors.
6. Minimalism is just another word for job aids.
7. Minimalism works only for simple domains.
8. Minimalism merely reflects the preconceptions of users.
9. Minimalism offers a complete documentation solution.
10. Minimalism has no theoretical foundation.

Fig. 1. Ten misconceptions about minimalism.

For example, it is not uncommon to see minimalism glossed as “cut text.” Brevity is a key element of minimalism, but only because it can facilitate task-oriented activity and learner-initiated reasoning, not as a self-sufficient end in itself. Wantonly slashing text and leaving other design characteristics unchanged will not lead to a minimalist design. It is easy to see the attractions of this misconception: random slashing is an inexpensive design transformation; if it really did produce better instruction and documentation, it would be the mother of all panaceas.

The first group of misconceptions we will discuss has this character: minimalism means the designer does not have to produce a complete design; minimalism means that all learning is learning by trial and error; minimalism means preventing learners from making errors; minimalism means that job aids can be employed as instructional manuals (see Fig. 1). They each exaggerate a single aspect of minimalism, often taking it more simply and more extremely than we intended. Unfortunately, these misconceptions are subsequently sometimes attributed to minimalism tout court (that is, instead of to a simplified and exaggerated caricature of minimalism). In essence, we will argue that a general view of minimalism cannot be reduced to any of these simplifications, that the effectiveness of the minimalist approach hinges on taking a more comprehensive, articulated, and artful approach to the design of information.

A second group of misconceptions devolves from the first group. These pertain more to the meta-strategy of minimalism than to its technical content. For example, because the original examples of minimalist documentation were developed in relatively simple application domains, people sometimes conclude that the approach is suitable only for such simple domains. Another example is that because the leading idea of minimalism is to build on the learner’s task-oriented motivation and reasoning, a critical requirement of minimalism is to fit the pre-theoretical expectations of users. Because minimalism describes an expansive program for the design

Manuscript received January 1996; revised February 1996.

J. M. Carroll is with the Department of Computer Science, Virginia Tech (VPI & SU), Blacksburg, VA 24061-0106 USA.

H. van der Meij is with the Department of Instructional Technology, Twente University, P.O. Box 217, 7500 AE Enschede, The Netherlands.

Publisher Item Identifier S 0361-1434(96)03978-1.

of documentation and training, people sometimes conclude that the intent of minimalism is to offer a comprehensive design model for all documentation. Because minimalism is a design philosophy, often conveyed or illustrated by sets of design heuristics (e.g., [2]), people sometimes conclude that minimalism comprises only heuristics without a systematic foundation in science.

The most serious consequence of these assorted misconceptions is that they may confuse or deter prospective researchers and practitioners. It is important therefore to consider how the misconceptions arise and to try to clarify them with more sound conceptions. This, in turn, can improve the relevance and substantiveness of subsequent discussions. In the balance of this paper we will consider the ten misconceptions enumerated in Fig. 1. We take them up in the order indicated.

1. MINIMALISM MEANS BREVITY

Brevity is clearly implied by the term “minimalism.” And brevity has always been a part of the minimalist approach to instruction and documentation. However, it has never been the totality of the approach. In some presentations, it is explicitly enumerated as one of several principles that work together to produce minimalist designs [2], [3]. In some presentations, it is identified as a derivative property of minimalist designs, caused by other, more central principles [1].

Nevertheless, it is typical to see mere brevity taken as the central thrust of minimalism. There are several apparent reasons for this. First, simplifying things in this way makes minimalism easier for commentators to explain and discuss. But it also yields a caricature of minimalism. For example, Hallgren [4] complains that Carroll, in his book *The Nurnberg Funnel*, “feels the need to make lengthy explanations to fellow practitioners on reducing their writing to the essentials” (p. 16). The irony in this appraisal seems witty, but in fact *The Nurnberg Funnel* is one presentation of minimalism that explicitly treats brevity as a derived property. The “lengthy explanations” Hallgren apparently did not view as worthy of analysis are in fact the principles of minimalism from which brevity derives.

A second possible reason for this misconception originates in the politics of the development process. There is still a substantial tendency for development managers to see instruction and documentation as a kind of tax on their budgets. Many of these managers may not care a whit about producing more effective training and documents; they just want to think less about supporting users, and spend less of their resources on it. They may wish to believe that designing and publishing fewer pages and panels will cost less. But this reasoning rests on serious fallacy. Minimalist design in documentation, as in architecture or music, requires identifying the core structures and content. It hinges on being able to make good decisions on what to do, say, or show, and on what *not* to include. This typically requires more skilled (expensive) developers and greater development effort.

The standard, systematic approach of responding to new design requirements with functions, new conditions, and new pages/panels of documentation will not lead to minimalist

results, but neither will the approach of slashing or simplifying. Developers must create genuine and intellectually engaging instructional opportunities for users. They must include just the right amount of information, not too much or too little. This typically involves *more* development effort, particularly when it is first adopted by a given designer or by an organization. It typically costs more, not less, as writers and developers need to reconsider some of most basic strategies and assumptions that guide their work [1], [5], [6].

Minimalism can be cost-effective for managers, but not because it makes documentation insignificant or inexpensive. Its focus on responding creatively to actual users and actual tasks entails a development process in which prototype documents are continually redesigned and tested. Feedback from users and their work-contexts pervades the process. A current-best prototype is always available to give managers a concrete status summary, enabling more fruitful discussions. The key savings in effort are to the user, not to the developer: minimalism significantly reduces training time while supporting better user performance. Experiments have shown reductions of between 25% and 40% training time [7]–[9].

2. MINIMALISM MEANS INCOMPLETE INSTRUCTIONAL ANALYSES

A key technique in minimalist documents is the use of incomplete descriptions and instructions. As in the case of brevity, this technique has always been presented as part of a more comprehensive set of techniques, not as a one-line design philosophy unto itself. And, also as in the case of brevity, this technique should be seen as the consequence of deeper principles: incomplete descriptions and instructions can pose problems to the learner, can suggest goals and activities, and can guide self-initiated investigation and discovery.

However, incompleteness has sometimes been misconceived by commentators and managers. The general misconception is that incomplete documentation is obtained through incomplete analyses. Of course, completeness is always a matter of degree: no document can be a complete description of a product or procedure. Writers and information developers always must find a balance in the detail they present. Creating effective but deliberately incomplete documentation requires some sort of theory of user inference. This is not a simple matter [10]. It hinges upon making just the right assumption about the prior knowledge and skills of the intended audience.

Fig. 2 presents an example of an instructional procedure for starting up a program [11], [12]. The procedure that users must follow involves changing the current directory, selecting a program, and completing a login/password dialog. Three steps are presented at a fairly high level of precision with respect to the task-element of making or changing the current directory, but the steps provide absolutely no guidance or even information about the login/password dialog or menu navigation actions that are also required in the situation.

This approach would be appropriate for users who were quite familiar with the login/password dialog and menu navigation elements but who did not know how to change the current directory. While it is possible that people like this could exist, it is unlikely: all three of these elements are

Chapter 1: How to start the program

You can start the program as follows:

1. After the 'C:\' type: cd\hours
2. Press: Enter
3. Choose: 'sawmenu'

The main menu now appears on the screen.

Fig. 2. An example of incomplete instructions from a manual developed by Glasbeek [11].

basic skills. In fact, two of five subjects tested in this study experienced considerable difficulties with the omitted topic of menu navigation. This kind of incompleteness is not guided by any theory or analysis of what the user knows or can infer. At best, it can convey an impression of sloppiness; at worst, it can leave users bewildered.

The attractiveness of this simplification of minimalism is analogous to the case of brevity. If documents could be made more incomplete by random omission of steps and conditions, and indeed thereby become more effective to users, it would be possible to realize huge savings in development and production costs. However, as the example in Fig. 2 shows, incomplete analyses of user knowledge, skills, and needs lead to poor documents, whether they are more or less complete.

As in the case of brevity, designing effective, incomplete instruction and documentation requires more, not less analysis. Determining what can be left out of a document involves analysis of users' prior knowledge and skills, task and needs analysis, and an empirical identification of likely hot-spots. Draper and Oatley [13] draw the critical distinction:

An instruction to "open the header and add a phrase" need not be expanded into "go to the Format menu, execute the Open Header command, click in the new window and type a phrase, close the window" if users can be relied on to search the menus for commands they do not know. . . . Such instructions are not designed by thoughtless omission, however. If users do not already know about opening headers, it is important that the phrase used ("open the header") be a close match to the command name they must search the menus for, or the search will be poorly constrained. (p. 228).

To implement the minimalist approach reliably it is necessary to carefully assess user knowledge and skill against the tasks users will be carrying out. We have found that a good approach to this is scenario analysis: creating and walking through scenarios of use, often in collaboration with real users [14]. In some cases, it may be useful to carry out structured task analyses or information processing analyses [8], [15], [16].

A particular focus of such scenario analyses must be potential sites of user error. Research consistently indicates that error-free performances of users are the exception rather than the rule [7], [17]–[23]. No matter how hard writers try, users are prone to make errors. Locating the hot spots in the program and manual is therefore a crucial part in the pre-writing analyses in a minimalist approach. Likewise, we believe that

Open a File (first presentation)

1. Press the Alt key
2. Press the \downarrow key
3. Press the Enter key

Open a File (second presentation)

1. Go to the menu bar
2. Press the \downarrow key
3. Press the Enter key

Open a File (third presentation)

1. Go to the menu bar and choose File
2. Press the Enter key

Open a File (fourth presentation)

1. Open file x by activating the menu and selecting Retrieve

Open a File (fifth presentation)

1. Open file x

Fig. 3. The use of a fading technique can reduce the number of mistakes and look backs.

writers should, at this stage, examine the best ways for dealing with the problems of users [1], [2], [23].

The problems users may have with the program and the manual are often found by logical analyses as well as by testing with the intended audience. After finding a problem, the writer must decide whether it should lead to changes in the program, to changes in the manual, or to the addition of error information [23]. Unfortunately, most task-analytic procedures ignore breakdowns or mistakes; "what if" scenarios are hardly ever analyzed. Such "what if" scenarios are an essential part of an instructional analysis in a minimalist approach.

People do not learn everything in one pass. Just as it can be wise to employ "progressive disclosure" in the number of menu options presented on screen to novice users, it can be wise to use fading as a way to gradually decrease the support for users. For example, when designing for an audience of adults with no computer knowledge, even simple tasks such as choosing from a menu may best be taught by gradually decreasing the support users are given. While this is not inconsistent with an approach in which every chapter can be worked through independently from other chapters, it surely is much more difficult to take this notion into account in designing a modular approach that works well for the user.

For instance, in an early version of our minimal manual for WordPerfect we asked users merely to achieve a given goal if the procedure for that goal had been presented in an earlier chapter. Testing revealed that this assumed too much learning, leading to mistakes and look backs by some of the users. We therefore decided to fade the information more gradually, adding reminders, but making successive reminders less complete (see Fig. 3). These later manuals were more successful in terms of reducing the number of mistakes and look backs.

Testing keeps one in touch with reality. As people become more and more experienced computer users, they can forget the difficulties they had as a novice. For example, Hayes (reported in [24]) showed that the more knowledge of a topic writers have, the poorer their predictions of difficulties for users.

3. MINIMALISM MEANS TRIAL-AND-ERROR LEARNING

Discovery learning is clearly a key element in the theoretical foundation of the minimalist design philosophy. Minimalism

assumes that learners should be active, that they should work on real or realistic tasks as they learn. Part of the reasoning for this is that real and realistic tasks are highly motivating [25]. Engaging in genuine activities during learning also better supports transfer to real situations by bringing the learning situations and transfer situations into closer correspondence [26]. In addition, solving real problems helps students become independent learners [27], [28].

It is, however, easy to confuse discovery learning with learning by trial and error [29]–[31]. As Williams and Farkas [32] put it, “we believe that the stated minimalist goal of enabling the learner to accomplish real work while learning a program is often thwarted by the act of compelling that learner to induce, *through trial and error*, the correct procedures needed to accomplish that work” (p. 41, emphasis added).

The confusion hinges on taking systematic (i.e., rote) curricula and trial-and-error as exhaustive possibilities, which they are not. Discovery learning is not equivalent to the absence of all curriculum and support; rather, it entails fundamentally different kinds of curriculum and support than passive and rote-structure approaches. This distinction has always been drawn in minimalism (see [1, pp. 104–110], describing research from 1981 contrasting guided exploration with learning by unguided immersion in a problem situation), but perhaps not clearly enough.

Effective discovery learning must be carefully supported: learners must have enough knowledge to form appropriate goals, pursue relevant activities, and draw correct conclusions. Implementing this requires knowledge analysis of what learners can be expected to do at various points throughout a discovery learning curriculum. Learners must be enabled to learn from mistakes and simultaneously be protected from distracting errors and confusions. Opportunities to discover must retain the motivating aspects of real tasks but also be tractable for learners.

A technique for ensuring that learners have the prerequisite knowledge to benefit from exploration, that they will want to explore, and that they will have a fair chance of success is to place invitations for exploration immediately after practice on a related task. This is illustrated in Fig. 4.

The “on your own” section in Fig. 4 does not merely ask students to explore. It cues them to consider what other goals they might want to pursue in relation to searching a text. The suggested goals for exploration are related conceptually and procedurally to the operations practiced in that context. This proximity helps to create what Bruner [28] called the “well-prepared mind”; the student has just been thinking about forward search, and thus is prepared to discover backward search. And the exploration is quite tractable: in this case, searching backward requires only two slightly different actions (i.e., position the cursor at another place, and select another option from the Search menu).

Our research indicates that preparing and cueing exploration strongly encourages learners to explore. For example, in one study, we found that such cues induced 81% more exploratory episodes than occurred with a control manual [34]. In another study, we found that 74% of all invitations to explore in a minimal manual for WordPerfect did lead to explorations by

Searching a Text

You can position the cursor quickly to a word or part of a sentence by searching for this text.

1. Position the cursor at the beginning of the file
2. Go to the menubar and press twice on the → key
3. Choose the command FORWARD and press the ENTER key

WordPerfect asks what you want to search for. Check to see if the prompt → **Search:** is on your screen.

4. Type any word(s) from the text
5. Press the F2 key

If the code **[Hrt]** appears after the word, you have pressed the ENTER key instead of the F2 key. Remove the code by pressing the BACKSPACE key. Then press F2 for searching the word(s).

WordPerfect automatically positions the cursor at the first occurrence of the word(s) in the text.

If you hear a bleep and the text ***Not found*** appears, you may have made a typing mistake. Try again.

On your own: Searching text

The commands NEXT and PREVIOUS enable you to find out if the word you have been searching can also be found elsewhere in the text. You can find these commands under the SEARCH option. Try them and see.

Fig. 4. An example of “on your own” section in a minimal manual developed by van der Meij and Lazonder [33].

the users. In contrast, the invitations to explore in a control (i.e., conventional) manual were significantly less tempting. Only 41% of all invitations were followed by explorations [35].

How much cueing is appropriate for effective exploration depends on the knowledge and skill of the learners. For some audiences, fewer cues may suffice, and more extensive (i.e., deeper) examinations into the menu options (e.g., “search and replace”) may be possible and challenging. The manual presented in Fig. 4 was intended for an adult vocational audience with no prior computer experience. Therefore, it has a fair degree of cueing in the “on your own” section.

The role of errors in learning is not ignored in conventional instructions and documentation but it is given too low a priority [36]. Errors play a distinctive and central role in approaches that incorporate discovery learning. Discovery learning tends to view error as a natural event in inquiry, a kind of event that should be put to use. Accordingly, a substantial amount of attention is directed at supporting the recognition and diagnosis of errors. At the same time it is also acknowledged that instructional environments must also

protect learners from making choices or performing actions that can severely disrupt task flow. One of the ways to create a fairly safe environment for exploration is to use a training wheels technology in which errors are identified to the user but their consequences are blocked [37]. This preserves the original system state and makes it easy for the learner to continue.

There are various ways to retain the motivating aspects of real tasks while reducing the complexity of reality. A prominent technique is to use sample files that enable users to create or revise a complete product without burdening them with all of the tasks that usually need to be performed for such a product. For example, users may be offered templates that help create a fully formatted text but that do not rely on having to specify all the codes for headings, letter types, margins, and so on. Another way of helping users cope with a complex and real task is to offer various information sources in the environment. For example, to assist Smalltalk programmers with a debugging task, we have combined passive help with a Commentator explaining the specific expressions used by the programmer, a Goal Poster offering a best guess analysis of the programmer's on-going activity, and a Guru providing strategic critique of the programmer's solution.

4. MINIMALISM DOES NOT SUPPORT PEOPLE WHO LEARN BY READING

Minimalism finds it important to value and support the reading strategies that people spontaneously apply. At the same time we also believe in Wright's [38] adage that "quality writing provokes quality reading." In other words, we try to find the right balance between accommodating to the users' reading strategies and using principles and techniques of good instructional design to evoke action-oriented reading.

The pervasive emphasis on learner activity in minimalism has led some to question whether minimalism is intended to serve learners who wish to read, or whether it can serve them. This is a more complex issue than it first appears; it involves several misconceptions about learners, about minimalism, and about alternatives to minimalism.

First, it is by no means clear that there are people who can learn skills merely by reading about them, without any active practice or performance of those skills or their components. Those who criticize minimalism for emphasizing learning activity, and thereby failing to support those who learn by reading, have not always been clear about what they take "learning by reading" to mean [12], [39]. They have not produced any compelling reasons or evidence to believe that skills can be learned by reading alone. They imply that minimalism emphasizes learning activity too much but have not been specific as to how much is enough.

Many reader-oriented alternatives to minimalism incorporate strong assumptions about reading. One of the assumptions is that users do as requested by the experimenter. Another is that users read everything before doing anything. These assumptions lead to certain design principles that writers should be very cautious in following when designing a manual whose usage is not artificially constrained. In our experience,

even learners who declare that they are the type of person who reads everything before trying to do anything, quickly and spontaneously begin interacting with the system as they read [40]. By way of illustration, we discuss the important studies of Charney, Reder, and Kusbit [41], and of Sweller and Chandler [42].

Sweller and Chandler [42] argue that some programs are hard to learn because they require the user to integrate information from different sources (e.g., manual, keyboard, and screen). The training materials they designed to support such integration were found to be superior to the ones that did not support these. We agree with the need to support integrative (in our term, coordinative) activities [3]. However, Sweller and Chandler were interested in learning by reading, and so the subjects were required to study the entire manual before they were permitted to practice with the system. Our view is that this assumption about reading is a severe and quite unnatural constraint, and that this is reflected in the manual they designed and studied.

Fig. 5 displays a page from the Sweller and Chandler manual presenting linkages among cursor location information in the drawing area, the status area, and the keyboard of a CAD/CAM system. This information is presented in a strictly descriptive style without reference to user tasks or activities. As a result, it does not capture and support the dynamic relations that constitute coordinative information.

Charney *et al.* [41] suggest that it can be advantageous for people to engage in problem solving (rather than merely executing prescribed instructions) when learning about a new program, and their research supported that notion. We agree with the need to have users engage in problem-solving activities such as inferencing and with the emphasis on devising one's own plan for achieving a task [1]. However, Charney *et al.* studied strictly sequential reading; they did not allow users to look backward or forward in the manual. Indeed, each piece of information appeared on its own separate page (see Fig. 6). This approach affords well-controlled laboratory experiments. However, its implications are unclear for circumstances in which manuals are *designed*, and not merely controlled, and for situations in which people are free to use materials as they wish (see [10, p. 81]).

A main idea in the design of our manuals has been to encourage sequential processing and to support a random access approach as well. One of the ways to realize this is by carefully structuring the manual around user scenarios. We follow user scenarios for structuring the content of our manuals. Typically, these scenarios lead first to the presentation of a few key tasks that are expanded upon later on. For example, in our text processing manuals the core is formed by a "create—save—print" scenario. After having acquired these basic skills, users are presented a number of chapters that belong to a "revise" scenario (e.g., chapters on locating text in a document, on moving text within a document, and on changing the presentation of words and of a document as a whole). But we do not expect users to process the manual in linear fashion. Indeed, such reading is quite uncommon; in our studies about 1 of every 20 subjects reads everything in this order. As a rule, users frequently look back to earlier sections

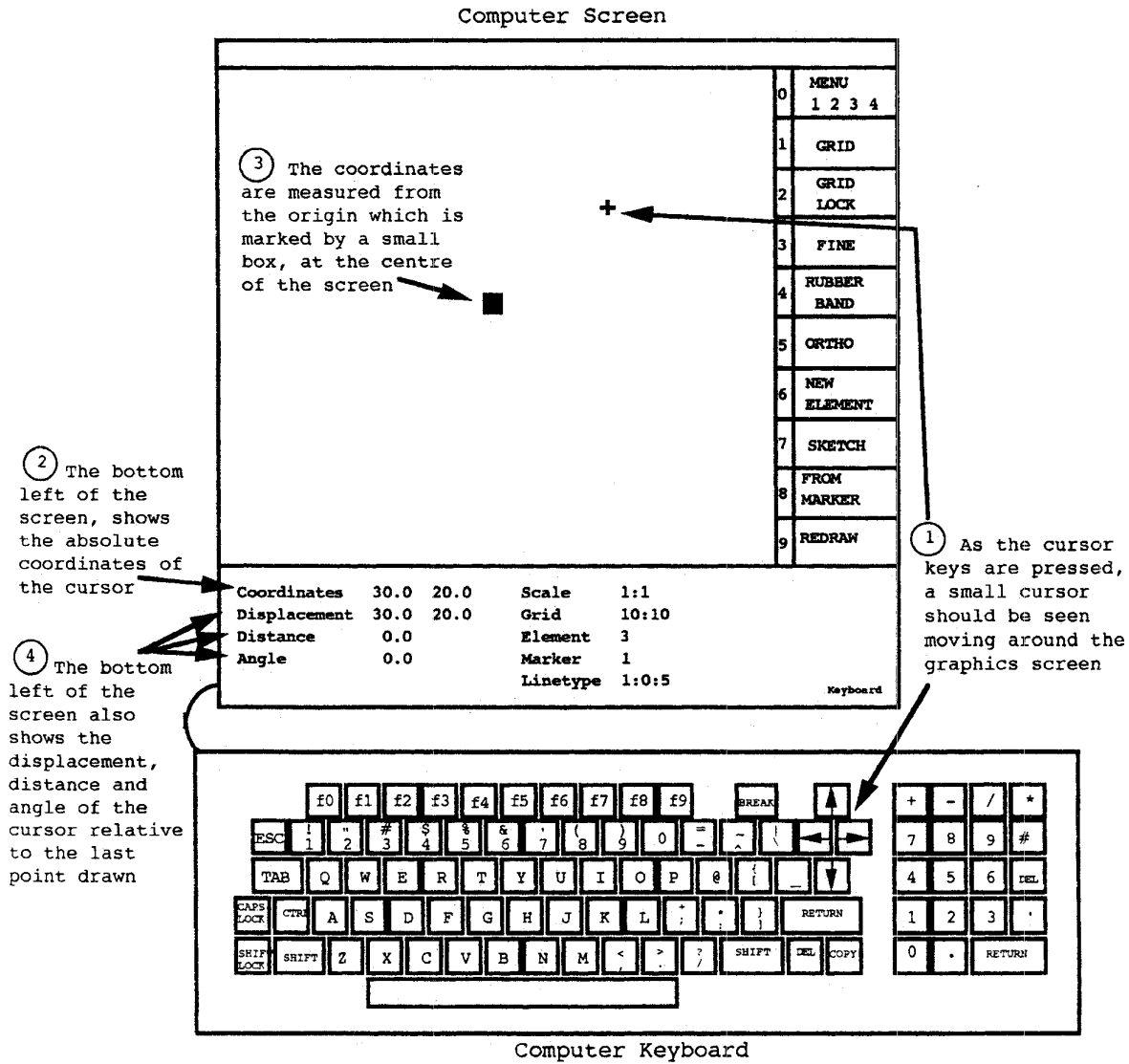


Fig. 5. A page from the CAD/CAM manual developed by Sweller and Chandler [42].

and chapters, peek at chapters and sections ahead, and redo chapters or tasks.

Our observations also reveal they do attend to most of the information presented in the minimal manual and that they process that information as intended. For example, 94% of all action steps were executed rather than read or skipped; 78% of learners checked the screen when prompted by coordinative information; 60% of learners who made an error used the error information in the manual to correct it.

When people are free to direct their own learning, motivation becomes a central issue. The first thing to do is to make a favorable initial impression. It is vitally important to avoid giving the manual a massive appearance [5], [43]. Once inside, the same impression must be sustained. This can be done by creating short chapters that take around thirty minutes to work through. Nothing motivates more than success (after effort). For this reason, we propose using real or realistic tasks as soon as possible, providing a safety net where needed,

and varying direct instructions with built-in problem-solving activities that offer a conquerable challenge. Interestingly, a fair number of these minimalist principles and heuristics are also mentioned as design suggestions that positively affect motivational aspects such as attention, relevance, confidence, and satisfaction. For example, Keller [25], [44] argues that real or realistic tasks are important for relevance and satisfaction, and that a safe trajectory of activities and challenging but conquerable tasks or problems are important for building confidence.

5. MINIMALISM OVEREMPHASIZES ERRORS

User errors are a major focus of minimalism. Most importantly, the mere fact that errors occur is acknowledged in minimalism. It is still quite typical for approaches to the design of instruction and documentation to ignore errors as a central phenomenon of using information, to implicitly assume that

(Page 10)
(DAB) To put the columns in order of ascending size, move the columns for Teaspoon and Tablespoon to the left and the column for Gallon to the right.

TYPE THIS: >F1 [RETURN]
 /M . B1 [RETURN]
 /M . C1 [RETURN]
 [-] [-] /M . G1 [RETURN]

(Page 9)
(CHI) Alphabetize the names by putting the rows containing Steele and Stewart further down in the appropriate spots. Start with cell A1 as the current cell.

TYPE THIS: /M . A7 [RETURN]
 /M . A7 [RETURN]

(Page 8)
The worksheet should look like this:

	A	B	C	D	E	F	G	H
1	1st Qtr	2nd Qtr	Jan	Feb	Mar	Apr	May	Jun
2	30	60	10	10	10	20	20	20

(Page 7)
(FIG) Move the columns for the quarterly totals together to the left of the worksheet.

TYPE THIS: >D1 [RETURN] /M . A1 [RETURN]
 >H1 [return] /M . B1 [RETURN]

(Page 6)
MOVE COLUMN OR ROW

The Move command moves the entire row or column that contains the current cell to another position on the worksheet.

PROCEDURES:

/M [FROM] . [TO] [RETURN] Moves the contents of row or column in the [FROM] coordinate to the row or column specified in the [TO] coordinate.

The Move command requires the following information:

- The FROM Coordinates: The coordinates of a cell in the row or column that you wish to move. VisiCalc automatically fills in the coordinates of the current cell (e.g., D5) as the FROM coordinates. If the current cell is not in the row or column you wish to move, type [BKSP] to erase these coordinates and type the coordinates of a cell in the row or column you want to move. Then type a period. Three periods appear on the edit line. Now you can type the "TO" coordinates.
- The TO Coordinates: The coordinates of a cell specifying the destination of the move. The TO coordinates must contain either the same column letter or the same row number as the FROM coordinates. The VisiCalc program determines whether to move a row or a column by comparing FROM and TO coordinates: if the column letter in the two coordinates is the same, then a row is moved; if the row number is the same, then a column is moved.
- The difference between the FROM and TO coordinates tells VisiCalc where to put the moved information. If the FROM coordinates (e.g., D5) have the same column letter as the TO coordinates (e.g., D3), then the contents of row 5 will move up to row 3. If the FROM coordinates (e.g., D5) have the same row number as the TO coordinates (e.g., B5), then the contents of column D will move left to column B.

VisiCalc makes room for the row or column you move by shifting the other rows and columns over. So moving a column or a row to a new location does not "cover up" any other entries.

Fig. 6. The full text from five consecutive pages of the problem-solving manual developed by Charney *et al.* [41].

errors do not occur [41, p. 324]. The facts go another way: error detection, diagnosis, and recovery consume somewhere between a quarter and half of the user's time ([19], [22], [23], [37], [45], and studies of Davis and Magers mentioned in [17]). Any realistic approach to instruction and documentation must focus substantial attention on errors. There are many tradeoffs and complexities; substantial design effort is always required. Recognizing and recovering from a particular error

might be a good pedagogical opportunity for the user. Another error might be a snakepit of side effects. Yet another might be pedagogically unproductive but difficult to block in the software. It might be best handled by a warning.

Some commentators have suggested that minimalism overemphasizes error and error support, making a fuss about something that “most of us would endorse as good, old-fashioned, common sense” [32, p. 41] and that “it is fairly common for manuals to anticipate users errors and to provide correctives for errors” [30, p. 186]. These assertions are wishful thinking. A recent survey on the presence and design of problem-solving information (i.e., error-information) in conventional manuals revealed that six of the eight manuals for a word processor failed to help users deal with a set of specific problems. In addition, an analysis of 60 conventional manuals in that same survey revealed that when there was problem-solving information, it was hard to find. For example, 33% of the tables of content and indexes in the manuals gave no references to problem-solving information, and 86% of the problem-solving information on the page was not marked in any way [36].

Other commentators have taken a single technique investigated in one or another minimalist project as a general approach to all error. For example, the training-wheels technique of blocking error consequences [37] has been critiqued in this way [46], [47]. Errors are not homogeneous. Some errors immediately intrigue learners; they wonder what caused the error; they want to replicate it, to analyze it, to try variations. These are good candidates for which to encourage error diagnosis and recovery. Other errors, just as clearly, annoy users from the first keystroke to the final recovery. When the frustration experienced by a learner obstructs the possible insights, the error recovery can no longer be seen as productive.

On the one hand, minimalism seeks to capitalize on error. Errors raise questions to users; they prepare the mind to draw new insights. For example, our MoleHill tutor for Smalltalk incorporated a tool we called the Guru, which provided a critique of the student’s strategy, delivered immediately after the completion of a project [48]. The critique focused on errors of programming and design style. For more serious errors, we have employed more directive interventions: warnings, checkpointing, and explicit error-recovery suggestions. We do not advocate structured practice with error handling to address potential negative effects of error (e.g., demotivation, anxiety, stress, and points of no return; see [46], [47]). Our view is that such training excessively compromises the goal of being task oriented. Moreover, the error-information in a minimal manual sufficiently deals with the negative effects of error.

On the other hand, minimalism seeks to provide a safety net for errors that are difficult, distracting, and/or unilluminating to recover from. For example, the training-wheels interface intercepted and blocked user actions that would have led to serious error consequences. Instead of allowing the error to occur, the system displayed a message saying that the selected function was not available in the training-level system. Such a message conveys a lot. It confirms for the user that the selected command was indeed selected, while at

the same time making it clear that the command was not executed by the system. Saying that the command is not available in the training-level system suggests that, in some other level of the system, the command will be available. Thus the user is encouraged to remember that function, to try it again later. But the user is spared the effort and potential complications of trying to recover from the error consequences that would have ensued if the command had been executed.

6. MINIMALISM IS JUST ANOTHER WORD FOR JOB AIDS

Some authors have argued that the essence of minimalism is merely an expanded role for job aids. For example, Horn [39, p. 9] says, “In one respect Carroll appears to reinvent the wheel: the job aid” (see also [29], [49, p. 18]). Gong and Elkerton [8] characterize minimalism as producing documentation “whose organization, down to the lowest level procedural step, communicates purely procedural or ‘how-to-do-it’ information” (p. 102).

Clearly, there is something to these remarks. Minimalism aims to get users started quickly on real tasks. To a considerable extent, this involves conveying basic procedural information very succinctly. Conceptual information is included only when necessary for adequate understanding and execution of tasks: the achievement of meaningful tasks is the foundation for subsequent conceptual elaboration.

Some of this is true of job aids: job aids are succinct procedures. Typically, however, they are used for reference by people who, at an earlier point in time, have received instruction [50]. Thus they are not designed to convey a model of the system or of the task domain. From the standpoint of learning operations and tasks, job aids are often an arbitrary set of cards that belong together only because they are packaged together. They are not designed to provoke task-oriented reasoning on the part of the user, but merely to support execution of a set of steps. They are intended merely to help people “play back” a set of instructions; they do not address errors.

One of the earliest minimalist projects developed and studied in the minimalist approach was also the one most similar in design to job aids. Guided Exploration cards were an early exploration of the amount of information and structure required by secretaries learning to use word-processing equipment [34]. Each card briefly addressed a particular functional goal that could be understood on the basis of their prior knowledge of office tasks (see Fig. 7). The cards were deliberately incomplete, often providing only hints, so that users would stay focused on the task. Each card separately addressed one basic task goal without reference to material covered on other cards; the set of cards was delivered as an unbound deck. Finally, each card included specific checkpoint information (to help users detect and diagnose errors) and error-recovery information (to help them get back on track).

Another example of minimalist documentation that initially seems much like a job aid comes from Gong and Elkerton’s study [8]. Fig. 8 illustrates a page of their procedural manual.

These two examples also illustrate several key differences between minimalist materials and job aids with respect to the

?

Typing something

▶ In the terminology of the computer, you will be "creating a document". Use the TASK SELECTION menu to tell the computer that you want to create a document.

▶ You can give your document *any name* you want, but you *cannot* use the same name for two different documents.

▶ You can begin to type when you see a typing page on the screen.

(Think of this display as a blank piece of paper but remember that you do not need to worry about margins or tabs)

▶ Press the big RET (carriage return) key to start a new line or to skip lines.

▶ When you are *done typing* or want to leave the typing page to do something else, you want (in terms of the computer) to "*end use*" of your document. Press REQST, type the word "end", and press ENTER.

✓ As you are typing, what you type will appear on the screen.

🚑 If you cannot get to the CREATE menu, press the ESC key.

🚑 You will see the TASK SELECTION menu appear and you can then try again.

🚑 Is the name of the document unique?

Fig. 7. An example of a guided-exploration card from the work of Lazonder [58].

type of support each provides to users. Guided Exploration cards only occasionally offer explicit prescriptions for procedures. Whenever possible, users are given instructions that are general enough to support user actions for related procedures or tasks. Figs. 7 and 8 illustrate this principle well. Users are not given exact prescriptions on what to do in typing or saving but are given enabling hints. Job aids, in contrast, succinctly present the step-by-step minutiae of procedures; as tools for learning and thinking, these are de-skilling characteristics, and not at all consistent with minimalism.

The Guided Exploration cards provide frequent feedback; each card includes at least one explicit coordinating anchor between the system state and the instructional material. This feedback conveys to users that they are on the right track; more broadly, it confirms the user's current understanding and helps to build self-confidence. Providing feedback also helps users coordinate and switch attention among the screen, the manual, and the keyboard. Finally, feedback gives users a point of reference when they do make an error; it indicates a place from which they might have gone wrong.

The mere presence of error-information in the Guided Exploration cards does not distinguish them from job aids. Some job aids are designed especially for trouble-shooting. What sets the two apart, however, is that the Guided Exploration cards integrate corrective with constructive information. Regular ("constructive") procedural instructions are accompanied by information about what to do if things go wrong. In most job aids, the support for constructive and corrective actions is hardly ever integrated. In contrast, minimalist instruction puts a high priority on putting the error-information where it is likely to be most helpful, which is in the immediate vicinity of error-prone actions [2].

By taxonomizing the procedures of a domain in terms of the basic task goals that users want to pursue, Guided Exploration cards convey a task-oriented model of the system and the domain of activity. Conveying such a conceptual framework is not an objective of job aids; it is sufficient for job aids to merely lay out the steps for various procedures. This focus on conveying goals and plans, and not just procedural steps, clearly differentiates all minimalist materials from job

TOPIC 5 - SAVING INFORMATION TO DISK STORAGE

1. Press the key labeled "Alt" and the "M" key at the same time.
2. Press the "S" key.
3. Create a file for the onscreen parameters and calculations:
 - A. Press the "D" key.
 - B. Press the "C" key.
 - C. Verify that the bottom window of the display looks like the following:

Respond to each prompt. <ESC> to delete, <CR> without entry to skip.

Please select the disk drive A, B, C, etc. and press <CR>, [.]

Give subdirectory if any: \.....\ Eight letter max then <CR>.

Give filename: [.....]. Eight letter max, no extension, then <CR>.

<CR> for default: (), <C>reate, <D>irectory, ESC quit.
 - D. Decide in which drive you want to save the information. Then type in the letter corresponding to that drive. Then press RETURN.
 - E. Decide if the file will be saved in a sub-directory. If so, then type in the name of the subdirectory and press RETURN. If not, then just press RETURN and proceed to the next step.
 - F. Type in the name of the file in which the information will be saved. Then press RETURN.
4. Decide if all the entered information (drive name, subdirectory name, and filename) are all correct. If so, press the "Y" key, and you have finished the save procedure. If not, press the "N" key and return to step 3C.

Fig. 8. A page from the procedural manual developed by Gong and Elkerton [8].

aids [13, see p. 228]. Indeed, subsequent minimalist projects emphasized this more than the Guided Exploration cards (among others, they lack goal descriptions), thereby drawing the distinction even more sharply.

7. MINIMALISM WORKS ONLY FOR SIMPLE DOMAINS

Minimalist instruction developed in relatively simple domains. Most of the early work addressed elementary word processing functions out of a practical convenience, and because this domain offered the chance for greater and more rapid practical impact. Some writers have mistakenly concluded from this that minimalism is intended *only* for simple domains, and that it can work only in such domains [4], [30], [39]. The reasoning seems to be that a simplified, action-oriented approach to elementary skills cannot in principle accommodate complex integrated system environments. In our opinion, there is no reason to believe that even the most intricate and complex domain cannot be conveyed through an action-oriented approach that focuses on core skills.

Some of the specific principles of minimalism too may have triggered the misconception. For example, the principle of getting started fast [1] could suggest that minimalism is applicable only to the earliest stages of learning. But in fact one can get started fast anywhere along a learning trajectory. And the minimalism claim is that this indeed is what learners prefer to do. Heuristics such as selecting or designing instructional activities that are real tasks, creating components of instruction that reflect the task structure, and providing closure of chapters support these action-oriented preferences [2]. Similarly, the

minimalist principle of exploiting prior knowledge [1] could suggest that minimalism is applicable only to domains people are already familiar with. But in fact, adult learners always have relevant prior knowledge and cannot help but engage it when they try to learn. The issue is not whether there is prior knowledge, but helping the learners to identify what prior knowledge is relevant, and how. Misinterpreting the principles along these lines could lead to the conclusion that minimalism is suitable only for instruction in simple (and known) domains and only for instructing the first-time user.

One of our main motivations in undertaking the development of minimalist instruction for Smalltalk and object-oriented design was to challenge the "simple domain" misconception. Our user group in that work was professional programmers at IBM. These highly skilled individuals wanted to continue to develop their skills by learning the object paradigm. We developed special programming environments [51], [52], and a minimalist tutorial [53], [54]. This was an extensive and successful minimalist project, carried out in an extremely complex, technical domain.

Another signal of the applicability of minimalism to a variety of domains and applications comes from the reported studies in the literature. Minimalist instruction has been produced for word processing (Displaywriter, WordPerfect), computer-assisted design, desktop publishing (PageMaker, ViewPoint), a time-registration program, telephony (Hayes Fax-modem, Norstar), an occupational safety application, e-mail (UNIX and VME), programming (CNC and Smalltalk), Hypercard, and database programs (see [1], [2]). And this variety is growing rapidly; we are constantly tracking down new developments.

It is true that most minimalist work has concentrated on instructing first-time users, but this does not indicate an inherent problem in applying it to advanced training or reference documentation applications (e.g., as suggested in [39]). In fact, two of the earliest commercial applications of minimalism were in the area of reference documentation—for the Xerox Viewpoint Electronic Publishing system and the Norstar telephone system (both reported in [1]). In addition, in some of our recent projects we obtained promising outcomes of minimalist reference materials. We do agree, however, that much work yet needs to be done on finding out how minimalism can contribute better reference documentation.

8. MINIMALISM MERELY REFLECTS THE PRECONCEPTIONS OF USERS

One of the key ideas of minimalism is to take advantage of the learner's task-oriented motivation and reasoning. People *want* to accomplish meaningful goals; they *want* to try things out, to learn by doing. This can be read as merely an injunction to satisfy every preference of the user. And indeed, there is a deep issue here about the appropriate role of the user in the design process [55]. Our intention is far narrower, however. Whether or not we take the design stance of satisfying every user preference, we should at least respect the integrity of the user's activity [2]. This is not a matter of giving the user 256 colors, animated graphics, and full-motion video. It is a matter of designing tools and information so that they do not obstruct the user's task-oriented goals. In this minimal sense, it is a matter of ensuring that a presumed tool is not antithetical to use.

This may seem obvious, but our early work showed that the dominant approaches to designing instructional material in fact did obstruct and impair task-oriented activity. One of the reasons that this situation persisted for quite some time is that people often think that what they are used to is also what is best. From previous schooling and experiences, most people are quite familiar with study books and so this is what they put up with. They (or managers) may simply voice social inertia, (e.g., "It worked in the past," "It is too costly to change"). In addition, lacking comparative materials, they may also have been unaware of better alternatives.

Our efforts to improve the situation began by observing scores of users. We carefully documented how initially well-motivated users were successively defeated by repetitive exercises and expansive conceptual sections, ultimately wondering aloud what they were supposed to be doing or learning [40], [56], [57]. This suggested the need for improvement. To convince technical writers (and managers) of the possibilities for improvement, we then began to design minimalist instructions and conducted a number of comparative experiments. These studies clearly showed the superiority of minimalist instructions over conventional instructional materials (see [1], [2], [7], [58]). In addition, users often indicated their satisfaction with these materials.

In our user-centered approach it has, however, never been our chief intention to "merely" please learners. Users do not have to love the manual in order to learn from it. They have

to accept it to the extent that they use it effectively; people's meta-knowledge about their own information needs is often flawed. Thus we have, over the years, encountered users who voice the desire for more complete manuals, but we have weighed this misfit between their desires with the results we have obtained for learning outcomes, namely, that they learned more than their peers who used more complete manuals.

At the same time, we have always found it necessary but difficult to create instructions that work best for the majority of users in a given situation and that simultaneously come closest to theoretically ideal instructions. This is a hard balancing act. For example, users often have different, or conflicting, opinions about the things they would like to see changed in instruction or information. One cannot design to suit user preferences in such cases, because one is pulled in opposing directions. So here theoretical arguments prevail. But while it may be theoretically ideal to have people read in advance, or solve problems before they engage in task executions [41], [42], this approach does not work for most users. So here practical considerations lead the way.

9. MINIMALISM OFFERS A COMPLETE DOCUMENTATION SOLUTION

Minimalism offers a simple but expansive program for the design of documentation and training. From this, one could conclude that minimalism constitutes a comprehensive design model for all documentation, a complete documentation solution. For example, Farkas and Williams [30] consider the minimal manual (and other single training tools) to be offered as complete solutions. A major point of their discussion is to contend that these minimalist approaches are not complete solutions; they directly contrast them with what they call a "contemporary documentation set (CDS)"—a tutorial, a user's guide, a quick-start manual.

The first point to make here is that there are potential downsides to having complete documentation solutions. People can get lost in the training and documentation options, as well as getting lost in any particular component. The positive effects of minimal manuals are telling in this respect.

In addition, we believe that a complete documentation solution requires an actively evolving solution, that it does not lie in a relatively stable and contemporary documentation set. Increased insights into the specific needs and deficiencies of users should prompt designers to create new solutions. For example, the contemporary documentation set has not been able to solve the problem of unused commands; users still typically use only a very small portion of the options of a program [59], [60]. Likewise, the set offers no solution for fixed but inefficient routines of users. For these (new) goals, new solutions are called for (e.g., some programs already offer advice to users about their inefficient routines).

Not only the goals of instruction are changing, the means of instruction are too. For example, the complete documentation solution of today is likely to be a mixture of paper and on-screen support. In our view, one of the most pertinent issues here is that of the interplay between the (software) product and the user support. We have directed some of our efforts

to this issue, such as with the creation of the Smalltalk Guru, MoleHill's intelligent tutoring system, and the training-wheels technology [1], [48], [61], [62]. But the topic is surely under-analyzed. For example, we have yet to see the first article outlining the main principles of the interplay between on-line help and paper support.

An altogether different reason that minimalism does not offer a complete documentation solution resembling the CDS of Farkas and Williams [30] is that it is an orientation toward design that does not aspire to provide a final cookbook. The relevant theory and practical experience about learning and using information that bear on the design of instruction and documentation are too vast and too diverse to be susceptible to cookbook treatment.

10. MINIMALISM HAS NO THEORETICAL FOUNDATION

Because the development of minimalism has been so firmly rooted in the world of design and design evaluation, some writers have come to see it as essentially consisting of rules-of-thumb, or to put it more grandly, as an "aesthetic" of documentation. They have accordingly complained that minimalism lacks a theoretical foundation, that even if it does work well, we will never know *why* [4, p. 12].

This is of more than merely "academic" interest: rules-of-thumb cannot support systematic, technical progress or refinement. Rules-of-thumb are grounded only vaguely and holistically in standards of practice. They are bound together only by being listed together. They do not support deductions or inferences. Basically, one takes them or leaves them. Thus if minimalism's foundations could not be identified and explicated, it would not, in principle, be capable of providing a framework for developing new approaches to instruction and documentation.

Two characteristics of minimalism, or more specifically of presentations of minimalism, may have encouraged this misconception. First, statements of the core principles of minimalism have varied over the years (contrast [1]–[3]). This might have led some readers into worrying that the principles are somewhat arbitrary, and that they therefore do not have a sound foundation. This is entirely mistaken. The content of the principles has in fact changed very little over the years. The varied presentations of minimalism reflect our best guesses as to the most effective way to articulate the essential content of the approach. Beyond this, we have always resisted taking ourselves so seriously as to sanctify a minimalist litany: our interest has always been in developing and evaluating ideas and techniques, not in mouthing particular words.

A second characteristic that may have encouraged the view that minimalism lacks theoretical foundations is that most developments and presentations of minimalism are strongly empirical. We do not find this problematic—the design of instruction and documentation is fundamentally empirical. Theoretical foundations are significant only if they have strong empirical content, that is, if they provide specific guidance to designers and are empirically testable. It is, unfortunately, typical to see information design philosophies justified by relatively vacuous appeals to intuition and assorted theory.

For example, Carroll [1] observed how Gagné and Briggs [63] was "justified" by broad appeals to behaviorist learning theory, while Gagné, Briggs, and Wagner [64] was "justified" by broad appeals to cognitive learning theory: these were successive editions of the same book, presenting the same design approach—only the foundation had changed! Readers looking for such "foundation" will have been frustrated by minimalism.

Indeed, the empirical nature of the minimalist approach has facilitated many focused empirical studies. Thus a considerable body of work replicates the results of Carroll *et al.* [7] with respect to minimal manuals: [6], [8], [9], [13], [58], [65]–[70]. Draper and Oatley [13] conclude their review with the statement that "minimalist instruction is a robust and reliable method for writing documentation that out performs most conventional manuals" (p. 222). A variety of empirical studies have examined specific aspects of the minimalist approach: task-orientation and error-information [8], [22], [23], [58], brevity and inferencing [65], the depiction of screens to facilitate user coordination [71], and various kinds of problem-solving practice [72].

The third characteristic of presentations of minimalism that may encourage the misconception that the approach lacks foundation is the unwavering incompleteness of these presentations: No definitive set of minimalist guidelines has ever been published; no step-by-step procedure for creating minimalist documentation exists (the closest approximation to this is [2]). Developers new to minimalism frequently complain about this; their expectations have been prefigured by prior experiences with systematic approaches that have the explicit goal of reducing information design to rule-following [4], [8], [58], [69], [73]–[75]. Minimalism has always had the goal of opening up documentation design as a process, while focusing it on the objective of supporting the user's sense-making:

What is instructional design know-how? Is it merely the ability to follow checklists? Is it merely to have a checklist? Or is it a fundamental commitment to a goal and to the process of discovery and invention that spring from having that goal? The key to minimalist instruction is not, for example, whether the designer blocks error consequences but whether the designer has made the far more fundamental commitment to discover how to support the learner's sense-making efforts. An ironic way to put this is that our goal is to fail to get designers merely to apply our guidelines but in doing so to succeed at getting them to discover minimalism and to invent minimalist approaches far more effective than those discussed here [1, pp. 304–305].

This misconception, in any case, is a misconception. *The Nurnberg Funnel* cites a foundation of over 200 books and technical papers drawn from educational psychology, learning theory, cognitive science, human–computer interaction, and information design. The exposition of the basic model [1, pp. 77–90] alone contains over 80 technical citations. The theoretical foundation, as we understand it, has three main roots: John Dewey's view [76] that the realm of the mind

includes the situations and tools that comprise problems and are used to solve them, Jean Piaget's view [77] of the mind as transforming itself by confronting and solving new sorts of problems, and, most directly, Jerome Bruner's view [27], [28] that students must be active in their own learning in order to become independent learners. This foundation has indeed become conceptually and empirically stronger through the course of the last decade, for example, in developments of constructivist theories of the mind [26].

At a finer level of articulation, minimalism rests on a broad foundation of specific psychological theories and results (e.g., action theory, ARCS-motivation theory, cognitive load theory, the principle of worked examples, the just-in-time principle, the redundancy effect, and the split attention effect). For example, a fair number of minimalist principles and heuristics are also presented as design suggestions in the ARCS-motivational theory of Keller [25], [44]. In addition, the task-oriented nature, the usage of real or realistic tasks, the support of coordinative actions, an experience-related deepening of skill, and a striving toward increasing independence of the user, all fit within the general framework of action theory [78], [79].

CONCEPTIONS OF MINIMALISM

Minimalism necessarily evolves with the context of information technology and information development practice. Action orientation in instruction and documentation, for example, may not yet be standard practice but is widely employed—in striking contrast to the context before the minimalism. Task orientation is also widely accepted, but more of a frontier for application, particularly in complex domains, for which it is often difficult to create meaningful tasks that are still manageable within relatively short spans of time. Support for error-handling and problem-solving is still not standard practice but has become accepted as “common sense.” Use orientation, support for the great variety of user activities and concerns, has in many cases become a team effort of graphic designers, instructional designers, technical communicators, system designers, and users themselves—directed at rich but invisible scaffolding, at helping more by imposing less.

We do not regard misconceptions of minimalism as merely mistakes. Although we have tried to investigate and articulate the minimalist approach clearly and effectively, we are well aware that much remains to be done. Misconceptions about minimalism—at the very least—embody feedback for those who wish to develop and communicate the minimalist approach. They are pointers to issues that have not yet been effectively clarified and resolved. In this sense, each of the ten misconceptions (and there may well be more than ten!) can be seen as an implicit need statement for further investigation and analysis.

It is in that spirit that we have tried here to identify several of these misconceptions, to indicate why and how we believe they might have arisen, and to address them, based on our own understanding of what minimalism is and of the current state of evidence and practice. Our objective is to stimulate more convergent, more productive discussion and development of minimalism. As we have tried to emphasize and epitomize in

this discussion, minimalism cannot be reduced to an operational dictum: it is a program for continuous discovery for both users and developers.

The rate and scope of change in the technology context of both users and developers presents more opportunities for discovering minimalism and more challenges for applying minimalism than ever before. We hope that readers will take these ten misconceptions about minimalism as ten directions for further research and analysis. We look forward to further lively discussion and investigation of minimalism emerging from this context.

REFERENCES

- [1] J. M. Carroll, *The Nurnberg Funnel: Designing Minimalist Instruction for Practical Computer Skill*. Cambridge, MA: M.I.T. Press, 1990.
- [2] H. Van der Meij and J. M. Carroll, “Principles and heuristics for designing minimalist instruction,” *Tech. Commun.*, vol. 42, no. 2, pp. 243–261, 1995.
- [3] J. M. Carroll, “Minimalist training,” *Datamation*, vol. 30, no. 18, pp. 125–136, Nov. 11, 1984.
- [4] C. Hallgren, “*The Nurnberg Funnel: A minimal collection*,” *J. Comput. Document.*, vol. 16, no. 1, pp. 11–17, 1992.
- [5] R. J. Brockmann, *Writing Better User Documentation: From Paper to Hypertext (Version 2)*. New York: Wiley, 1990.
- [6] G. VanderLinden, T. G. Cocklin, and M. McKita, “Testing and developing minimalist tutorials: A case history,” in *Proc. 35th Int. Technical Communications Conf.* Washington, DC: Soc. Tech. Commun., 1988, pp. RET 196–199.
- [7] J. M. Carroll, P. S. Smith-Kerker, J. R. Ford, and S. A. Mazur-Rimet, “The minimal manual,” *Human-Computer Interaction*, vol. 3, pp. 123–153, 1987/1988.
- [8] R. Gong and J. Elkerton, “Designing minimal documentation using a GOMS model: A usability evaluation of an engineering approach,” in *Proc. CHI'90 Conf.: Human Factors in Computing Systems*, J. C. Chew and J. Whiteside, Eds. New York: ACM, 1990, pp. 99–106.
- [9] A. W. Lazonder and H. Van der Meij, “The minimal manual: Is less really more?,” *Int. J. Man Mach. Studies*, vol. 39, pp. 729–752, 1993.
- [10] B. Mirel, S. Feinberg, and L. Allmendinger, “Designing manuals for active learning styles,” *Tech. Commun.*, vol. 38, no. 1, pp. 75–87, 1991.
- [11] H. Glasbeek, “Leerstijlen van computergebruikers,” (Learning styles of computer users), Masters Thesis, Utrecht Univ., Utrecht, The Netherlands, 1992.
- [12] ———, “Improving the quality of tutorials: Does minimalism really help?,” in *Quality of Technical Documentation*, M. Steehouder, C. Jansen, P. Van der Poort, and R. Verheijen, Eds. Amsterdam, The Netherlands: Rodopi, 1994, pp. 77–83.
- [13] S. W. Draper and K. Oatley, “Action centered manuals or minimalist instruction? Alternative theories for Carroll's minimal manuals,” in *Computers and Writing: State of the Art*, P. Holt and N. Williams, Eds. Oxford, UK: Intellect Books, 1990, pp. 222–243.
- [14] J. M. Carroll, Ed., *Scenario-Based Design: Envisioning Work and Technology in System Development*. New York: Wiley, 1995.
- [15] D. H. Jonassen, W. H. Hannum, and M. Tessmer, *Handbook of Task Analysis Procedures*. New York: Praeger, 1989.
- [16] P. F. Merrill, “Job and task analysis,” in *Instructional Technology: Foundations*, R. M. Gagné, Ed. Hillsdale, NJ: Erlbaum, 1987, pp. 141–174.
- [17] B. Arnold and R. Roe, “User errors in human-computer interaction,” in *Psychological Issues of Human Computer Interaction in the Work Place*, M. Frese, E. Ulich, and W. Dzida, Eds. Amsterdam, The Netherlands: Elsevier, 1987, pp. 203–22.
- [18] R. W. Bailey, *Human Error in Computer Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1983.
- [19] S. K. Card, T. P. Moran, and A. Newell, *The Psychology of Human-Computer Interaction*. Hillsdale, NJ: Erlbaum, 1983.
- [20] ———, “The keystroke-level model for user performance time with interactive systems,” in *Human-Computer Interaction: Selected Readings*, J. Precece and L. Keller, Eds. Hemel Hempstead: Prentice Hall, pp. 327–356, 1990.
- [21] A. C. Graesser and K. Murray, “A question-answering methodology for exploring a user's acquisition and knowledge of a computer environment,” in *Cognition, Computing and Cooperation*, S. P. Robertson,

- W. Zachary, and J. B. Black, Eds. Norwood, NJ: Ablex, 1990, pp. 237-267.
- [22] A. W. Lazonder and H. Van der Meij, "The effect of error-information in tutorial documentation," *Interacting with Computers*, vol. 6, no. 1, pp. 23-40, 1994.
- [23] ———, "Error-information in tutorial documentation: Supporting users' errors to facilitate initial skill learning," *Int. J. Human-Computer Studies*, vol. 42, pp. 185-206, 1995.
- [24] P. Wright, "Issues of content and presentation in document design," in *Handbook of Human-Computer Interaction*, M. Helander, Ed. Amsterdam, The Netherlands: Elsevier, 1988, pp. 629-652.
- [25] J. M. Keller, "Development and use of the ARCS model of instructional design," *J. Instruct. Devel.*, vol. 10, no. 3, pp. 2-10, 1987.
- [26] T. M. Duffy and D. H. Jonassen, *Constructivism and the Technology of Instruction: A Conversation*. Hillsdale, NJ: Erlbaum, 1992.
- [27] J. S. Bruner, *Toward a Theory of Instruction*. Cambridge, MA: Harvard Univ. Press, 1966.
- [28] ———, *Beyond the Information Given*. New York: Norton, 1973.
- [29] M. J. DeWeaver and J. M. Bauman, "The Nurnberg Funnel: Designing minimalist instruction for practical computer skill," *Perform. & Instruct.*, pp. 25-26, Apr. 1992.
- [30] D. K. Farkas and T. R. Williams, "John Carroll's *The Nurnberg Funnel* and minimalist documentation," *IEEE Trans. Prof. Commun.*, vol. 33, no. 4, pp. 182-187, 1990.
- [31] A. L. Kamouri, J. Kamouri, and K. H. Smith, "Training by exploration: Facilitating the transfer of procedural knowledge through analogical reasoning," *Int. J. Man Machine Studies*, vol. 24, pp. 171-192, 1986.
- [32] T. R. Williams and D. K. Farkas, "Minimalism reconsidered: Should we design documentation for exploratory learning?" *SIGCHI Bull.*, vol. 24, no. 2, pp. 41-50, 1992.
- [33] H. Van der Meij and A. W. Lazonder, "An assessment of the minimalist approach to computer user documentation," *Interacting with Computers*, vol. 5, no. 4, pp. 355-370, 1993.
- [34] J. M. Carroll, R. L. Mack, C. H. Lewis, N. L. Grischkowsky, and S. P. Robertson, "Exploring a word processor," *Human-Computer Interaction*, vol. 1, pp. 283-307, 1985.
- [35] H. Van der Meij, "Catching the user in the act," in *Quality of Technical Documentation*, P. van der Poort, M. Verheyen, M. W. Stehouder, and C. Jansen, Eds. Amsterdam, The Netherlands: Rodopi, 1994, pp. 201-210.
- [36] ———, "Helpt de handleiding?" (Does the manual help?), *Tijdschrift voor Taalbeheersing*, vol. 17, no. 4, pp. 267-288, 1995.
- [37] J. M. Carroll and C. Carrithers, "Blocking learner errors in a training wheels system," *Human Factors*, vol. 26, no. 4, pp. 377-389, 1984.
- [38] P. Wright, "Quality or usability? Quality writing provokes quality reading," in *Quality of Technical Documentation*, P. van der Poort, M. Verheyen, M. W. Stehouder, and C. Jansen, Eds. Amsterdam, The Netherlands: Rodopi, 1994, pp. 7-38.
- [39] R. E. Horn, "Commentary on *The Nurnberg Funnel*," *J. Comput. Document.*, vol. 16, no. 1, pp. 3-11, 1992.
- [40] J. M. Carroll and S. A. Mazur, "LisaLearning," *IEEE Computer*, vol. 19, no. 11, pp. 35-49, 1986.
- [41] D. H. Charney, L. M. Reder, and G. W. Kusbit, "Goal setting and procedure selection in acquiring computer skills: A comparison of tutorials, problem solving and learner exploration," *Cognition and Instruction*, vol. 7, no. 4, pp. 323-342, 1990.
- [42] J. Sweller and P. Chandler, "Why some material is difficult to learn," *Cognition and Instruction*, vol. 12, no. 3, pp. 185-233, 1994.
- [43] L. Thing, "What the well-dressed manual is wearing today," *Tech. Commun.*, vol. 31, no. 3, pp. 8-12, 1984.
- [44] J. M. Keller, "Motivational design of instruction," in *Instructional-Design Theories and Models: An Overview of Their Current Status*, C. M. Reigeluth, Ed. Hillsdale, NJ: Erlbaum, 1983, pp. 383-434.
- [45] J. Prümper, D. Zapf, F. C. Brodbeck, and M. Frese, "Some surprising differences between novice and expert errors in computerized office work," *Behav. Inform. Technol.*, vol. 11, no. 6, pp. 319-328, 1992.
- [46] M. Frese and A. Altmann, "The treatment of errors in learning and training," in *Developing Skills With Information Technology*, L. Bainbridge and S. A. R. Quintanilla, Eds. Chichester, UK: Wiley, 1989, pp. 65-86.
- [47] M. Frese, F. Brodbeck, T. Heinbokel, C. Mooser, E. Schleiffenbaum, and P. Thiemann, "Errors in training computer skills: On the positive function of errors," *Human Computer Interaction*, vol. 6, pp. 77-93, 1991.
- [48] S. R. Alpert, M. K. Singley, and J. M. Carroll, "Multiple multimodal mentors: Delivering computer-based instruction via specialized anthropomorphic advisors," *Behav. Inform. Technol.*, vol. 14, pp. 69-79, 1995.
- [49] P. A. Carlson, "From document to knowledge base: Intelligent hypertext as minimalist instruction," *J. Comput. Document.*, vol. 16, no. 1, pp. 17-31, 1992.
- [50] A. Rossett and J. Gautier-Downes, *A Handbook of Job Aids*. San Diego, CA: Pfeiffer, 1991.
- [51] J. M. Carroll, J. A. Singer, R. K. E. Bellamy, and S. R. Alpert, "A view matcher for learning Smalltalk," in *Proc. CHI'90 Conf.: Human Factors in Computing Systems*, J. C. Chew and J. Whiteside, Eds. (Seattle, WA, Apr. 1-5, 1990). New York: ACM, pp. 431-437.
- [52] M. B. Rosson, J. M. Carroll, and C. Sweeney, "A view matcher for reusing smalltalk classes," in *Proc. CHI'91 Conf.: Human Factors in Computing Systems*, S. P. Robertson, G. M. Olson, and J. S. Olson, Eds. (New Orleans, LA, Apr. 27-May 2, 1991). New York: ACM, pp. 277-283.
- [53] J. M. Carroll and M. B. Rosson, "Managing evaluation goals for training," *Commun. ACM*, vol. 38, no. 7, pp. 40-48, 1995.
- [54] M. B. Rosson, J. M. Carroll, and R. K. E. Bellamy, "Smalltalk scaffolding: A minimalist curriculum," in *Proc. CHI'90 Conf.: Human Factors in Computing Systems*, J. C. Chew and J. Whiteside, Eds. (Seattle, WA, Apr. 1-5, 1990). New York: ACM, pp. 423-429.
- [55] J. Greenbaum and M. Kyng, Eds., *Design at Work: Cooperative Design of Computer Systems*. Hillsdale, NJ: Erlbaum, 1991.
- [56] J. M. Carroll and R. L. Mack, "Learning to use a word processor: By doing, by thinking and by knowing," in *Human Factors in Computer Systems*, J. C. Thomas and M. L. Schneider, Eds. Norwood, NJ: Ablex, 1984, pp. 13-51.
- [57] R. L. Mack, C. H. Lewis, and J. M. Carroll, "Learning to use office systems: Problems and prospects," *ACM Trans. Office Inform. Syst.*, vol. 1, pp. 254-27, 1983.
- [58] A. W. Lazonder, "Minimalist computer documentation. A study on constructive and corrective skills development," Doctoral dissertation, Twente University, Enschede, The Netherlands, 1994.
- [59] M. B. Rosson, "Effects of experience on learning, using, and evaluating a text-editor," *Human Factors*, vol. 26, pp. 463-475, 1984.
- [60] P. Westendorp, *Gebruiksaanwijzingen (User Manual)*. Eindhoven, The Netherlands: Instituut Vervolgopleidingen, Tech. Univ. Eindhoven, 1993.
- [61] M. B. Rosson, J. M. Carroll, R. K. E. Bellamy, S. R. Alpert, and J. A. Singer, *MITTS: Minimalist Tutorial and Tools for Smalltalk/V*. Yorktown Heights, NY: Comp. Sci. Dept., IBM T. J. Watson Research Center, 1993.
- [62] M. K. Singley and J. M. Carroll, "Minimalist planning tools in an instructional system for Smalltalk," in *Proc. 3rd IFIP Conf. on Human-Computer Interaction Interact'90*, D. Diaper, D. Gilmore, G. Cockton, and B. Shackel, Eds. (Cambridge, UK: Aug. 27-31, 1990). Amsterdam, The Netherlands: North-Holland, pp. 937-944.
- [63] R. M. Gagné and L. J. Briggs, *Principles of Instructional Design*. New York: Holt, Rinehart and Winston, 1979.
- [64] R. M. Gagné, L. J. Briggs, and W. Wagner, *Principles of Instructional Design*, 3rd ed. New York: Holt, Rinehart and Winston, 1988.
- [65] J. B. Black, J. M. Carroll, and S. M. McGuigan, "What kind of minimal instruction manual is the most effective?" in *Proc. CHI+GI'87 Human Factors in Computing Systems and Graphics Interface*, J. M. Carroll and P. P. Tanner, Eds. (Toronto, Ont., Canada, Apr. 5-9, 1987). New York: ACM, pp. 159-162.
- [66] M. Frese, K. Albrecht, A. Altmann, J. Lang, P. von Papstein, R. Peyerl, J. Pruemper, H. Schulte-Goecking, I. Wankmueller, and R. Wendel, "The effect of an active development of the mental model in the training process: Experimental results in a word processing system," *Behav. Inform. Technol.*, vol. 7, pp. 295-304, 1988.
- [67] L. Olfman and R. P. Bostrom, "The influence of training on use of end-user software," in *Proc. Conf. on Office Information Systems*. New York: ACM, 1988, pp. 110-118.
- [68] J. E. Ramsay and K. Oatley, "Designing minimalist tutorials from scratch," *Instruct. Sci.*, vol. 21, pp. 85-99, 1992.
- [69] H. Van der Meij, "A critical assessment of the minimalist approach to documentation," in *Proc. 10th Annual Int. Conf. on Systems Documentation (SIGDOC'92)* (Ottawa, Ont., Canada, Oct. 1992). New York: ACM, pp. 7-17.
- [70] R. Wendel and M. Frese, "Developing exploratory strategies in training: The general approach and a specific example for manual use," in *Proc. 2nd IFIP Conf. on Human-Computer Interaction*, H. J. Bullinger, B. Shackel, and K. Kornwachs, Eds. Amsterdam, The Netherlands: Elsevier, 1987, pp. 943-948.
- [71] R. H. Nowaczyk and E. C. James, "Applying minimal manual principles for documentation of graphical user interfaces," *J. Tech. Writing and Commun.*, vol. 23, no. 4, pp. 379-388, 1993.
- [72] S. Wiedenbeck, P. L. Zila, and D. S. McConnell, "End-user training: An empirical study comparing on-line practice methods," in *Proc. CHI'95 Conf.: Human Factors in Computing Systems* (Denver, CO, May 7-11, 1995). New York: AC, pp. 74-81.

- [73] C. J. M. Janssen, "Research in technical documentation in The Netherlands," *Tech. Commun.*, vol. 41, no. 2, pp. 234–239, 1994.
- [74] R. S. Nickerson, "A minimalist approach to the 'paradox of sense making'," *Educ. Researcher*, vol. 20, no. 9, pp. 24–26, 1991.
- [75] S. D. Tripp, "Book review of: *The Nurnberg Funnel*," *Educ. Technol. Res. Devel.*, vol. 38, no. 3, pp. 87–90, 1990.
- [76] J. Dewey, *How We Think*. New York: Heath, 1910.
- [77] J. Piaget, *The Origins of Intelligence in Children*. New York: Norton, 1963.
- [78] D. A. Norman, "Cognitive engineering," in *User Centered System Design: New Perspectives on Human-Computer Interaction*, D. A. Norman and S. W. Draper, Eds. Hillsdale, NJ: Erlbaum, 1986, pp. 31–61.
- [79] K. Rieder and R. Oesterreich, "The task-oriented training methodology for training software packages," in *Atrium in Europe: The Development of Learning Abilities in Youth Training*, J. M. M. van der Sanden and F. J. J. van Bussel, Eds., Tilburg Univ., 1994.

Hans van der Meij is a researcher at the Faculty for Educational Sciences and Technology at Twente University, Enschede, The Netherlands. His main fields of study are questioning behavior and technical documentation. In the latter area, he has started his work by examining minimalism. And, while continuing his work on minimalism, he is now also examining the state of the art in technical documentation in a number of inventories. These inventories focus on issues like warnings, introductions, action commands, the accessibility of problem-solving information, and the combination of on-paper and on-line support. They are an attempt to link theory (and empirical research) with practice and thereby advance technical writing as an applied science.

John M. Carroll (SM'90) is Professor of Computer Science and Psychology, and Head of the Computer Science Department at Virginia Tech., Blacksburg. His research is in the analysis of human learning and problem-solving in human-computer interaction contexts, and in the design of methods, tools, and environments for instruction and design. His most recent books are *Scenario-Based Design: Envisioning Work and Technology in System Design* (New York: Wiley, 1995), and *Design Rationale: Concepts, Techniques, and Use* (with T. P. Moran), (Hillsdale, NJ: Erlbaum, 1996).