ASSESSING THE USABILITY OF A USER INTERFACE STANDARD

Henrik Thovtrup
Dialog Data
Jernbanevej 3H, 4.th.
DK-2800 Lyngby
Denmark

and

Jakob Nielsen
Bellcore
445 South Street
Morristown, NJ 07962-1910
USA

Correspondence should be sent to Jakob Nielsen, email: nielsen@bellcore.com

ABSTRACT

User interface standards can be hard to use for developers. In a laboratory experiment, 26 students achieved only 71% compliance with a two page standard; many violations were due to influence from previous experience with non-standard systems. In a study of a real company's standard, developers were only able to find 4 of 12 actual deviations in a sample system, and three real products broke between 7 and 12 of the 22 mandatory rules in the standard. Designers were found to rely heavily on the examples in the standard and their experience with other user interfaces.

KEYWORDS: Standards, Consistency, Examples, System Development, Developers, Attitudes.

INTRODUCTION

User interface standards have become the object of increasingly intense activities in recent years [Abernethy 1988; Holdaway and Bevan 1989], including work in the International Standards Organization (ISO) [Brooke et al. 1990] and the European Community [Stewart 1990]. Work is also going on in national standards organizations [Dzida 1989] and in several major computer companies [Berry 1988; Nielsen 1989b]. These activities are part of a general current interest in information processing standards [Berg and Schumny 1990] but are also based on the widely held feeling that consistency is one of the most important usability considerations [Nielsen 1989b]. Even though consistency is obviously not the only usability factor [Grudin 1989], there are still good reasons to strive to obtain it in balance with other usability considerations [Nielsen 1990b] in a usability engineering process [Nielsen 1991], and such additional considerations are indeed also included in many current standards activities.

Given the potential future importance of usability standards, it seems reasonable to study the usability of the standards themselves to assess whether developers can actually apply the content of the documents. Not much research is available on this topic yet, but existing evidence does indicate the potential for "meta-usability problems" (usability problems in a usability document). Mosier and Smith [1986] report that only 58% of the users of a large collection of interface guidelines found the information they were looking for (an additional 36% "sometimes found it"). de Souza and Bevan [1990] had three designers design an interface using a draft of the ISO standard for menu interfaces and report that they violated 11% of the rules and had difficulties in interpreting 30% of the rules. The draft standard was improved after the experiment, so the main lesson from this study is the need for usability testing of usability standards.

For a user interface standard to increase usability in the resulting products, two conditions have to be met: The standard must specify a usable interface, and the standard must be usable by developers so that they actually build the interface according to the specifications. As reported by Potter et al. [1990], a user interface may have usability problems even when an interface standard is followed without violations. The usability of the resulting interfaces is obviously extremely important for the development of the actual content of interface standards, but the present paper will concentrate on whether developers can use standards. First, we report on a small laboratory study, and the main part of the paper than reports on a field study of the use of a real standard.

LABORATORY STUDY OF A STANDARD

In one small experiment, 26 computer science students who were taking a user interface design class were asked to design an interface for a hypothetical company having a two page user interface standard. The standard described the use of several special function keys and the way the screen was partitioned into various fixed fields. In addition to the two page standard, the students were given a three page specification of a sample system that complied with the standard. The average degree of compliance in the resulting designs was rated at 71% by (subjectively) comparing them with a checklist of design elements specified in the standard.

These designers were presumably highly motivated to follow the standard since a part of their course grade was determined by their designs and they knew from previous lectures that compliance was to be considered a major usability consideration. Also, the standard was very small and easy to follow, making it almost impossible to overlook rules. Even so, the results show that the designs deviated substantially from the standard. The use of special keys deviated the most, possibly because the test standard specified a drastically different use of the keyboard than that used in most actual systems with which the students were familiar. For example, it specified the use of special "yes" and "no" keys to answer questions and did not use the traditional numbered function keys. It seemed that real, live systems influenced many of the designers more than the standards document. In fact, only 35% of the designs received a perfect score for compliance with the standard's use of special keys. The remaining 65% were at least partly influenced by the use of special keys in outside systems, and 15% of the designs were scored as having zero compliance on this point.

THE DATA COMPANY AND ITS STANDARD

We studied the actual use of an in-house user interface standard at a medium sized Danish company which we will call *The Data Company*. This company is a mixture of a software house and a service bureau and supplies several software products to its customers. The software is mainly mainframe software for traditional, text-only terminals, and each system typically contains between 50 and 300 screens. Because of this fairly large number of screens in each product, consistency is a highly desirable usability attribute for The Data Company. Also, several customers use more than one product, further indicating the need for an interface standard. The Data Company uses incompatible mainframes from two major computer companies with one extremely large vendor supplying most of the machines and another very large vendor supplying a smaller number of machines. A single interface standard was also desirable as a way to smooth over the differences between these two environments.

Standard or Guideline	*Pages*
DIN 66234 part 8 standard [1988] (note: *dense* pages)	6
The Data Company's standard	57
Apple Human Interface Guidelines [1987]	166
Motif™ style guide [OSF 1990]	167
"Advanced CUA" (for graphical interfaces) [IBM 1990a]	209
"Basic CUA" (for traditional terminals) [IBM 1990b]	288
Original CUA [IBM 1987]	340
OPEN LOOK™ [Sun Microsystems 1990]	404
Smith and Mosier [1986] guidelines	485
ISO 9241 (mostly not published yet)—*estimated*	532

Table 1. *Pagecount of various user interface standards or guidelines (including front matter). Note that ISO 9241 will eventually have 19 parts.*[1] *The pagecount given here is a projection on the basis of drafts of six of these parts, assuming that the remaining parts will have the same average pagecount as these parts. Also note that the complete set of ISO standards applicable to user interface design will include several additional documents such as the JTC1/SC18/WG9 work on "Specifications for objects, actions, operations, and applications" (50 page draft), "Cursor control" (13 page draft), and "Icons for use on screens" (34 page draft). See [Billingsley 1990a,b] for further lists of international standards.*

The Data Company released its interface standard in 1989 as a fairly small document of 57 pages (cf. Table 1). The standard was deliberately limited to dealing with text-only mainframe interfaces. Current plans call for the release of an additional standard for graphical interfaces in 1991.

After a small section giving recommendations for the process of constructing usable interfaces, the standard defines requirements for menu and command oriented systems. It defines standard function key assignments (F1=HELP, etc.) and a standard terminology (for example, "personal code" is to be used instead of terms like "userid"). The standard then describes methods to ensure that the user can control and customize the system (e.g., it must be possible to turn audible beeps off). A large part of the standard is devoted to a definition of proper screen design and layout, including the way input and output fields should be shown and labelled. Figure 1 shows an

[1] The 19 parts of ISO 9241 *("Ergonomic Requirements for Office Work with Visual Display Terminals")* are: 1: General introduction, 2: Guidance on task requirements, 3: Visual display requirements, 4: Keyboard requirements, 5: Workstation layout and postural requirements, 6: Environmental requirements, 7: Display requirements with reflections, 8: Requirements for displayed colors, 9: Requirements for non-keyboard input devices, 10: Dialogue principles, 11: Usability statements, 12: Presentation of information, 13: User guidance, 14: Menu dialogues, 15: Command dialogues, 16: Direct manipulation dialogues, 17: Form filling dialogues, 18: Question and answer dialogues, and 19: Natural language dialogues. The projection in Table 1 was calculated on the basis of available drafts of Parts 1, 2, 8, 10, 11, and 14.

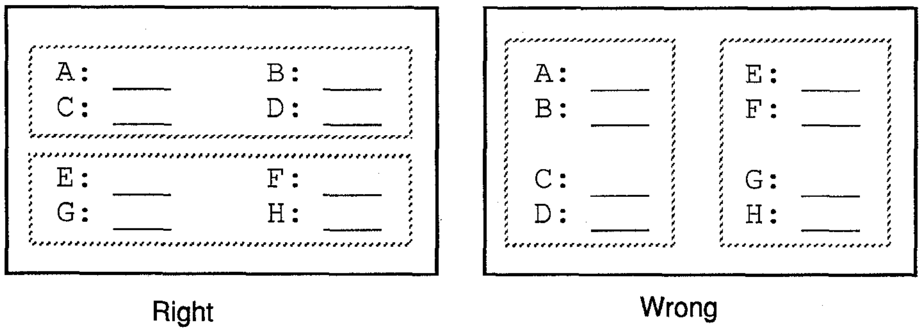


Figure 1. *Example from The Data Company's standard of how to position input fields to utilize the left-to-right tabulating sequence of cursor movements on alphanumeric terminals.*

example used to illustrate rules about the tabulator sequencing used to move the cursor between fields. The standard further contains sections on color and highlighting, printouts, help and error messages, and security. The standard has several example screen dumps used to illustrate the rules. These examples are in Danish and are mostly difficult to translate. Figure 2 shows a translated version of one of the examples used in the standard to illustrate the navigation system through hierarchical menus. This example only shows the header area of the standard screen layout whereas most examples show full screen dumps and actual text (menus, field labels, sample user input, etc.).

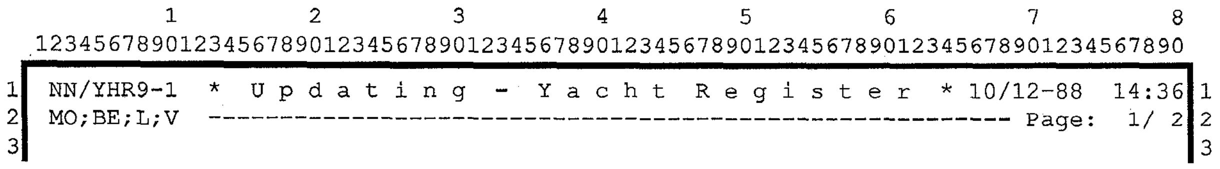
A mixture of methods was used to assess the usability of the standard. One study was aimed at The Data Company's developers and their opinions about the standard and their ability to follow it. Another study looked at The Data Company's actual products and how closely they followed the standard.

DEVELOPERS' SUBJECTIVE VIEW OF THE STANDARD

For the study of the developers, 15 developers from a total of seven different projects were visited and asked a number of questions using both a structured and a free-form interview. The participants were, unfortunately, selected with a bias towards developers with a higher than average interest in usability since participation in the study was voluntary. Indeed, 14 of the 15 participants indicated that they had taken a course in usability. This bias should be kept in mind in interpreting the results from the study.

In general, the developers had a very positive attitude toward the standard. As many as 73% could "fully agree" that The Data Company should have a user interface standard and nobody disagreed. The average agreement on a 1–5 scale was 4.7 (where 1="completely disagree" and 5="fully agree"). The developers also liked the actual content of the standard; the screen designs were rated 4.3 and the dialogue flow was rated 4.2 on a 1–5 scale with nobody using the negative ratings of 1 or 2. The developers even had a fairly positive opinion of the possibility for introducing an official national Danish standard to cover all vendors and software houses, giving a proposal to do so a rating of 3.6 on the 1–5 scale. When asked whether it would have been better for The Data Company to have adopted the standard from its main vendor instead of writing its own, the developers rated their agreement as 3.1 on the average on the 1–5 scale. The underlying distribution of attitudes towards vendor standards was bimodal, however, with a majority (53%) of developers having a neutral or partly positive attitude and a minority of 27% having strong negative feelings. In free-form interviews, some developers who developed for the minority vendor's machines claimed that the main vendor's standard was inappropriate for their platform because of differences such as varying keyboard design.

On the 1–5 scale, the developers only gave a rating of 1.9 to the suggestion that a set of recommendations would have been better than an actual standard, thus again expressing their favorable attitudes toward having standards. They did like the fact that the standard included some recommendations and "good advice" in addition to the formal requirements (rating this 4.7 on the 1–5 scale). During the free-form interview, one major reason mentioned for wanting a formal standard was that it helped minimize wasted time during project meetings. Prior to the introduction of the standard, a lot of time was spent arguing



```
             1         2         3         4         5         6         7         8
   12345678901234567890123456789012345678901234567890123456789012345678901234567890
 1  NN/YHR9-1  *  U p d a t i n g  -  Y a c h t  R e g i s t e r  * 10/12-88  14:36 1
 2  MO;BE;L;V  -------------------------------------------------------- Page:  1/ 2 2
 3                                                                                  3
```

Figure 2. *Example (translated) from The Data Company's standard of the header area of a screen with the user's navigational location encoded in the upper left corner (the user is in the YHR system's MO area's BE subsystem's L subsubsystem's V screen.*

about minor interface design details[2] whereas now it was possible to close such discussions rapidly by referring to the standard, thus making it possible to concentrate on higher-level matters.

Despite the mainly positive attitudes towards the standard, there were also negative points mentioned during the interviews. Even though 67% of the developers felt that the standard made it easier for them to design screens and develop the associated software, 20% disagreed with that statement. As many as 53% of the developers complained that they did not have sufficiently good programming tools to support the user interface requirements made by the standard.

The question about whether they would comply with the standard under all circumstances divided the developers into two camps, with 33% who would sometimes deviate from the standard. Of the 67% "loyalists," only 27% were "blind loyalists" who would follow the standard no matter what, whereas the other 40% could only partly agree that they would follow the standard under all circumstances. Considering the probable sampling bias of using volunteer developers with an interest in usability, it is likely that the true proportion of "deviators" is even larger for the non-sampled group of developers. One perspective on this result might be that a standard is only a true standard if it is followed closely under all circumstances[3] and that one should therefore institute very strict quality assurance policies to enforce compliance. An alternative perspective, as argued by Tognazzini [1989], is that user interface standards have a somewhat different character than other computer standard and one should actually be allowed to deviate from them if one has very good reasons for doing so. In any case, the replies to this question certainly indicate the need for some way to minimize deviations since the developers do not feel much bound by the standard.

One main underlying reason for the tendency to deviate from the standard can be found in the answers to the question whether the standard restricted the creativity of system designers. All "loyalists" answered no to this question while all "deviators" answered yes, suggesting that better compliance with standards will follow if they can be made to seem less confining. Again, all 33% "deviators" answered yes to whether the standard precluded new design ideas, further indicating a rationale for their discontent. Even so, 20% of the 67% "loyalists" also answered yes to this question, and the average "loyalist" rating of agreement was 2.8 on the 1–5 scale (only a very slight disagreement—in fact, almost neutral). This indicates that limitations on the product may not be as influential as are limitations on the people as a motivation to break the standard.[4]

Looking at the usability of the standard as a document, as many as 53% of the developers said that the rules in the standard were difficult to remember. Only 27% found the rules easy to remember, and 20% answered "don't know" to this question. 27% said that the standard was difficult to apply, 40% that it was easy to apply, and 33% had a neutral opinion on this issue.

Regarding the size of the document, 60% felt that the standard was too small, compared with 40% who felt that it was the appropriate size and nobody who felt that it was too large. As shown by Table 1, The Data Company's standard is indeed considerably smaller than most other standards. It does not follow, however, that a larger standard would actually be easier to apply or to remember. When it comes to traditional computer manuals, it is certainly often the case that users express a desire for larger and more complete manuals [Nielsen 1989a] even though smaller and more concise manuals are sometimes better for them [Carroll 1990].

MEASURING DEVELOPERS' ABILITY TO USE THE STANDARD

As reported above, many developers reported that the standard was difficult to remember and to apply. To measure the developers' actual abilities to use the standard, the same 15 developers were asked to apply the standard in a small test using a method similar to that used by Molich and Nielsen [1990]: Developers were presented with a concrete design in the form of screen dumps and were asked to list all its deviations from the standard. They were given 15 minutes to do so and were allowed to use the standard document as much as they wanted during the exercise. The test design contained four screens and represented a hypothetical system for the administration of a dog owners' tax. The design contained 12 deviations from the standard.

On the average, the developers only found 4.0 out of these 12 deviations. The top scorer was one developer who found 7 of the 12 deviations. This performance was surprisingly poor, especially considering the bias inherent in our sample of developers with above-average interest in usability. Table 2 lists the 12 deviations and the number of developers who found them. Three of the 15 developers had served as user interface coordinators for their projects. They found on average 5.7 as opposed to the 3.4 deviations

[2] It seems that *everybody* has an opinion about what function keys to use and what words to use in a menu [Furnas et al. 1987]. Debates on such issues can go on for hours and often do not lead to better designs if they are based on opinions rather than fundamental usability considerations.

[3] This is certainly true of almost all other standards used in the computing field.

[4] Developers may be allowed additional flexibility under the "conditional standards" approach [Williams 1989] where they can choose which of several rules are most appropriate to their unique circumstances. See also Scapin's [1990] premise-criteria-conclusion rule-based approach to reasoning about interface design.

Deviation from the standard	Developers finding the deviation
1. Absence of the colon which is supposed to mark the start of a data display field	15
2. No abbreviation given for menu options	13
3. Wrong format for a date (hyphen used instead of slash as a separator between elements)	9
4. Deviation from standard terminology ("personal identifier" instead of the proper "personal code")	5
5. F2 used for "main menu" command instead of F12	4
6. One screen did not have the required indication of navigational location in the menu hierarchy	4
7. The sub-system for updating the database did not contain an explicit update command	4
8. No measurement unit shown for the field giving the weight of the dog	2
9. Grouping of fields not appropriate for the tabulator sequence on the screen	2
10. Wrong format for a date (no leading zero given for month numbers <10 on the first screen)	1
11. The sub-system for updating the database did not contain an undo function	1
12. One of the menus has no Quit command (it should be possible to quit directly from everywhere)	0

Table 2. *Deviations from the standard in a four-screen test design. The deviations are sorted according to the number of the 15 developers who found each deviation.*

found by those 12 developers who had not had the role of user interface coordinator. This difference (significant at $p<0.01$) might be due to the coordinators' greater experience at looking at other people's screen design, but they still performed surprisingly poorly on the test task.

In addition to being poor at finding the actual deviations in the test screens, many developers claimed that certain design details were deviations even though they were not. Only two of the 15 developers did not indicate such spurious deviations. The average number of spurious deviations per developer was 1.6. The two most common spurious deviations were the centering of the line listing the valid function keys (mentioned by 11) and the use of navigational abbreviations on the screens instead of full subsystem names (mentioned by 7). The standard actually does not state anything about the justification of the line listing the function keys but all the example screenshots in the document happen to show left justified lines. As will be further discussed below, the examples were more influential than the actual formal specifications in the standard. The use of navigational abbreviations was authorized by the standard but had not been used by any actual system at The Data Company yet. It seems that concrete, implemented product designs were more salient than the abstract standards document in defining the corporate interface style in the minds of the developers.

Table 3 shows the distribution of the developers' use of the standards document during the experiment. They made comparatively little use of the main document containing the actual specification and rules of the standard and instead relied on the examples and the concrete lists of approved function keys and screen terms. This corresponds well with results from studies of written instructions in general [LeFevre and Dixon 1986] showing that users often rely the most on examples. Also, when asked during the interview, 80% of the developers could "partly agree" that the examples were used more than the actual rules and specifications. The remaining 20% would "neither agree nor disagree" with this statement.

PRODUCT COMPLIANCE

Three of The Data Company's released products were carefully inspected by the authors to assess their compliance with the standard. The developers of the system were confronted with the list of violations and in no case disputed they were indeed in conflict with the standard. In order to carry out the inspection, it was necessary to develop a standardized checklist listing all the rules in the standard and dividing them into three groups: Mandatory rules, voluntary rules, and guidelines for the development process (as opposed to rules for the resulting product). The checklist had 22 mandatory rules, 6 voluntary rules, and 5 guidelines for the design process. Table 4 shows the results from this compliance test. Most of the deviations were judged to have fairly minor impact on the final usability of the product and mainly have the effect of reducing the feeling of "product family" across products.

Of the five guidelines for the development process (study the characteristics of the user population, have user representatives participate during the process, run thinking aloud tests of the interface, use iterative techniques such as prototyping, and have an appointed coordinator for the entire user interface), all three projects had omitted the thinking aloud test and had complied with the remaining four.

With respect to the mandatory rules, Table 4 shows a fairly

Part of the document	% use
Main document (the actual standard specification)	25%
Example screenshots	21%
List of function keys and their standard use	21%
Table of contents and index	18%
Word list of approved screen terms	16%

Table 3. *Use of different parts of the standard during the experiment where 15 developers tried to find deviations in a sample design. The right column indicates the proportion of the total number of times the developers accessed the document.*

large number of deviations from the standard, especially with respect to the number of rules broken (between a third and a half of the 22 rules). The present data are too limited for any firm conclusions but seem to indicate a possible trend towards having more rules broken in larger interfaces. Given that a larger system (especially one with more screens) presents more opportunities for the designers to make mistakes, it is somewhat surprising that this trend is not stronger.

When asked why their products deviated from the standard, the developers mostly claimed that they had chosen alternative design solutions because they had found them to be better than the one mandated by the standard. Two other frequent explanations were that the development tools did not allow compliance with the standard and that the developers had indeed planned compliance but had not yet had time to implement it. The first explanation is related to the "loyalist"/"deviator" discussion above, and the latter two explanations indicate the need for good development tools to make it easy to implement interfaces that follow the standard [Tognazzini 1989]. Further explanations were that the developers were not aware of the rule they had broken or that they had overlooked the deviation. In no case did it turn out that the developers had actively misinterpreted the standard and designed a specific deviating interface feature in the explicit belief that they were following a rule from the standard. This indicates that the individual parts of the standard are reasonably understandable—perhaps because the standard almost always contains elaborations of the rules and backs them up with a rationale.

CONCLUSIONS

The results presented above show that user interface standards are very likely to be violated. Even our highly motivated, biased sample of developers did not express completely blind loyalty towards the standard. They were not able to apply the standard very well, and the concrete designs we investigated did indeed contain many violations.

To increase the usability of user interface standards, we recommend

- having development tools that support implementation of interfaces that follow the standard
- including concrete examples of correctly designed interfaces.

These recommendations correspond fairly well with the advice Happ and Cohen [1989] derived from interviewing a group of developers. We would add a warning to make sure that the examples do not inadvertently misrepresent the standard since the examples can have as much impact on developers as the formal specifications. Similarly, one should be aware that existing systems also influence developers heavily. Therefore, if a deviation from previous design practice is desired, it will probably be necessary to state so explicitly and to provide a rationale for why the new interface style is better.

Furthermore, the standards document itself should be designed according to recognized principles for good document design and include good access mechanisms such as a thorough index and table of contents, word lists, glossaries, and other checklists (such as a list of function keys). Table 3 indicates heavy reliance on such access mechanisms. Possibly computerized access mechanism such as hypertext [Nielsen 1990a, especially pp. 50–52] could be used to good effect, especially since they would allow the display of dynamic, animated examples which will probably be important for the standardization of modern interaction techniques.

Finally, our personal experience from this work indicates that a checklist of specified design elements and rules helps tremendously during a conformance quality assurance review.

Acknowledgements

The authors would like to thank Rita Bush, Susan Dumais, Tom Landauer, Jim Williams, and the anonymous *CHI'91* referees for helpful comments on earlier versions of the manuscript.

Disclaimer

It is the policy of Bellcore to avoid any statements of comparative analysis or evaluation of products or vendors. Any mention of products or vendors in this paper is done where necessary for the sake of scientific accuracy and precision, or for background information to a point of technology analysis, or to provide an example of a technology for illustrative purposes, and should not be construed as either positive or negative commentary on that product or that vendor. Neither the inclusion of a product or a vendor in this paper, nor the omission of a product or a vendor, should be interpreted as indicating a position or opinion of that product or vendor on the part of the authors or of Bellcore.

System	*1*	*2*	*3*
Number of screens	250	80	40
Number of developers on project	32	5	4
Deviations from mandatory rules	31	38	18
Number of mandatory rules broken (of 22)	12	9	7
Number of voluntary rules broken (of 6)	3	4	3
Number of design process guidelines broken (of 5)	1	1	1

Table 4. *Results from investigating three systems' compliance with the standard.*

REFERENCES

Abernethy, C.N. (1988). Human–computer interface standards: Origins, organizations and comment. In Oborne, D.J. (Ed.), *International Review of Ergonomics* **2**, 31–54.

Apple Computer (1987). *Human Interface Guidelines: The Apple Desktop Interface,* Addison-Wesley.

Berg, J.L., and Schumny, H. (Eds.) (1990). *An Analysis of the Information Technology Standardization Process,* Elsevier Science Publishers, Amsterdam, the Netherlands.

Berry, R.E. (1988). Common User Access—A consistent and usable human–computer interface for the SAA environment, *IBM Systems Journal* **27**, 3, 281–300.

Billingsley, P. (1990a). The standards factor: Committee updates, *ACM SIGCHI Bulletin* **21**, 4 (April), 16–19.

Billingsley, P. (1990b). The standards factor: Standards on the horizon, *ACM SIGCHI Bulletin* **22**, 2 (October), 10–12.

Brooke, J., Bevan, N., Brigham, F., Harker, S., and Youmans, D. (1990). Usability assurance and standardization—work in progress in ISO, *Proceedings IFIP INTERACT'90* (Cambridge, U.K., 27–31 August), 357–361.

Carroll, J.M. (1990). *The Nurnberg Funnel: Designing Minimalist Instruction for Practical Computer Skill,* The MIT Press.

de Souza, F., and Bevan, N. (1990). The use of guidelines in menu interface design, *Proceedings IFIP INTERACT'90* (Cambridge, U.K., 27–31 August), 435–440.

DIN (1988). *Bildschirmarbeitsplätze: Grundsätze ergonomischer Dialoggestaltung* (VDU work stations: Principles of ergonomic dialogue design, in German), Deutsches Institut für Normung **DIN 66234, Teil 8.**

Dzida, W. (1989). The development of ergonomic standards, *ACM SIGCHI Bulletin* **20**, 3 (January), 35–43.

Furnas, G.W., Landauer, T.K., Gomez, L.M., and Dumais, S.T. (1987). The vocabulary problem in human–system communication, *Communications of the ACM* **30**, 11 (November), 964–971.

Grudin, J. (1989). The case against user interface consistency, *Communications of the ACM* **32**, 10 (October), 1164–1173.

Happ, A.J., and Cohen, K.C. (1989). Consistency in the user interface: Common User Access impact study, *Technical Report* **TR54.513**, IBM Entry Systems Division, Boca Raton, FL.

Holdaway, K., and Bevan, N. (1989). User system interaction standards, *Computer Communications* **12**, 2 (April), 97–102.

IBM (1987). *Systems Application Architecture: Common User Access: Panel Design and User Interaction,* Document **SC26-4351-0.**

IBM (1990a). *Systems Application Architecture: Common User Access: Advanced Interface Design Guide,* Document **SC26-4582-0.**

IBM (1990b). *Systems Application Architecture: Common User Access: Basic Interface Design Guide,* Document **SC26-4583-0.**

LeFevre, J-A., and Dixon, P. (1986). Do written instructions need examples?, *Cognition and Instruction* **3**, 1, 1–30.

Molich, R., and Nielsen, J. (1990). Improving a human-computer dialogue, *Communications of the ACM* **33**, 3 (March), 338–348.

Mosier, J.N., and Smith, S.L. (1986). Application of guidelines for designing user interface software, *Behaviour and Information Technology* **5**, 1 (January–March), 39–46.

Nielsen, J. (1989a). What do users really want?, *International Journal of Human–Computer Interaction* **1**, 2, 137–147.

Nielsen, J. (Ed.) (1989b). *Coordinating User Interfaces for Consistency,* Academic Press, San Diego, CA.

Nielsen, J. (1990a). *Hypertext and Hypermedia,* Academic Press, San Diego, CA.

Nielsen, J. (1990b). Traditional dialogue design applied to modern user interfaces, *Communications of the ACM* **33**, 10 (October), 109–118.

Nielsen, J. (1991). *Usability Engineering,* Academic Press, San Diego, CA.

Open Software Foundation (1990). *OSF/Motif Style Guide,* Prentice Hall.

Potter, S.S., Cook, R.I., Woods, D.D., and McDonald, J.S. (1990). The role of human factors guidelines in designing usable systems: A case study of operating room equipment, *Proc. Human Factors Society 34th Annual Meeting* (Orlando, FL, 8–12 October), 392–395.

Scapin, D.L. (1990). Organizing human factors knowledge for the evaluation and design of interfaces, *International Journal of Human–Computer Interaction* **2**, 3, 203–229.

Smith, S.L., and Mosier, J.N. (1986). *Design Guidelines for Designing User Interface Software.* Technical Report MTR-10090 (August), The MITRE Corporation, Bedford, MA 01730, USA.

Stewart, T. (1990). SIOIS—Standard interfaces or interface standards, *Proceedings IFIP INTERACT'90* (Cambridge, U.K., 27–31 August), xxix–xxxiv.

Sun Microsystems (1990). *OPEN LOOK Graphical User Interface Application Style Guidelines,* Addison-Wesley.

Tognazzini, B. (1989). Achieving consistency for the Macintosh. In Nielsen, J. (Ed.), *Coordinating User Interfaces for Consistency,* Academic Press, 57–73.

Williams, J.R. (1989). Guidelines for dialogue design. In Salvendy, G., and Smith, M.J. (Eds.), *Designing and Using Human–Computer Interfaces and Knowledge Based Systems,* Elsevier Science Publishers, 589–596.