in: Workshop on Emerging Technologies in Human Engineering Testing and Evaluation.
NATO Research Study Group 24, Human Engineering Testing and Evaluation, Brussels 1997

FIFTH SESSION
Assessment of User Acceptance

ABOUT A METHOD TO MEASURE THE ERGONOMIC QUALITY OF USER INTERFACES IN A TASK INDEPENDENT WAY

Matthias Rauterberg

Human-Machine Interaction Group (MMI)
Institute for Hygiene and Applied Physiology
Swiss Federal Institute of Technology (ETH)
Clausiusstr. 25, CH-8092 Zurich, Switzerland
Tel: +41-1-632 7082, Fax: +41-1-632 1173
URL: http://www.ifap.bepr.ethz.ch/~rauterberg

Abstract

There currently are several views on human computer interaction in measuring interactive qualities of usability attributes: (1) the interaction-oriented view, (2) the user-oriented view, (3) the product-oriented view and (4) the formal view. Two different possibilities of measurement within the product-oriented view are introduced in this paper. Different types of user interfaces can be described and differentiated by the concept of "interaction points". Regarding to the interactive semantic of "functional interaction points" (FIPs), four different types of FIPs must be discriminated. Based on the concept of FIPs, the dimensions "[visual] feedback" and "interactive directness" can be quantified. Both metrics are helpful to classify the most common user interfaces: command, menu, and direct manipulation. The classification can be validated with the outcomes of several empirical comparison studies.

Keywords: User-interfaces, utility functions, testability, quantification, metrics

1 Introduction

The main problems of standards (ISO, DIN, etc.) in the context of software ergonomics is that they cannot measure user interface attributes in a quantitative and task independent way. Four different views on human computer interaction to measure interactive qualities currently exists (see also Rengger, 1991; Bevan, Kirakowski and Maissel, 1991, p. 651).

The *formal view*: usability is formalised and simulated in terms of mental models (formal concepts). Karat (1988) describes formal methods in the context of "theory-based" evaluation.

The *user-oriented view*: usability is measured in terms of the mental effort and attitude of the user ("questionnaires" and "interviews", see Kirakowski and Corbett, 1993).

The *product-oriented view*: usability is measured in terms of the ergonomic attributes of the product (quantitative measures). All heuristic evaluations (cf. Jeffries and Desurvire, 1992) carried out by ergonomic experts investigating a concrete product fall in this category, too.

The *interaction-oriented view*: usability is measured in terms of how the user interacts with the product ("usability testing"). This view is the most common one. All kinds of usability testing with "real" users are subsumed in this category (Kirakowski and Corbett, 1990).

The interactive qualities of user interfaces are currently quantified in the context of *interaction-oriented view* and *user-oriented view*, but these both approaches are time consuming and more or less expensive (see Figure 1). Usability testing is constrained to the investigated task solving processes and the selected users, too. On the other side, usability testing is characterized by a maximum of ecological validity. To cut down testing costs, it would be really helpful if usability attributes could be quantified in such a way that the extent of each attribute could be measured in--task independent--product features of the interface itself. This product oriented view can be differentiated in three approaches: (1) usability inspection methods (e.g., heuristic evaluation, see Nielsen and Mack 1994), (2) checklists (e.g., TCO 1992). In this paper we present an abstract concept to describe usability attributes of the most common user interfaces in a unique and pure quantitative form.

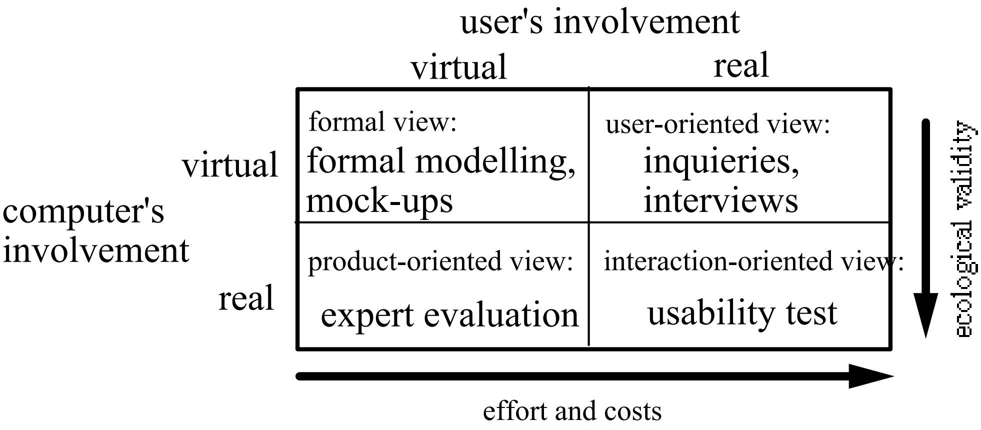


Figure 1. A classification schema of existing methods to measure usability aspects.

2 A Quantitative Description based on Interaction Points

It is necessary to define measures of usability for the product-oriented view, a concept of descriptive terms, which can be counted. The granularity of the descriptive terms must be on a medium level – not too specific (e.g. "push button", "menu option", etc.) and not too general (e.g. "transparent", "flexible", etc.). A level, at which it is possible to describe the different types of user interfaces ("batch", "command", "menu", "desktop") in a uniform and precise way, and at the same time a level is required that is powerful enough and easy to apply. The interaction space (IS) consists of two different interlaced spaces: the object space (OS), and the function space (FS). OS encloses all perceptible represented objects (PO) and all hidden objects (HO), which users can grasp and bring into the actual dialog context. The same situation is valid for FS: We have to distinguish between perceptible represented functions (PF) and hidden functions (HF). A concrete dialog context (DC) contains a subset of {OS ∪ FS}.

Table 1. The interaction space (IS) consists of the object (OS) and the function (FS) space

$IS := OS \times FS$	[interaction space]
$DC \in IS$	[dialog context]
$OS := PO \cup HO$	[object space]
$FS := PF \cup HF$	[function space]
$PO := PDO \cup PAO$	[(perceptible) representations of objects]
$HO := HDO \cup HAO$	[hidden objects]
$PF := PDFIP \cup PAFIP$	[(perceptible) representations of functions]
$HF := HDFIP \cup HAFIP$	[hidden functions]
$PDFIP := \{(df, pf) \in HDFIP \times PF: pf = \delta(df)\}$	[(perceptible) represented DFIP]
$PAFIP := \{(af, pf) \in HAFIP \times PF: pf = \alpha(af)\}$	[(perceptible) represented AFIP]
$FIP := DFIP \cup AFIP$	[interaction-points]
$DFIP := PDFIP \cup HDFIP$	[FIPs of dialog functions]
$AFIP := PAFIP \cup HAFIP$	[FIPs of application functions]
δ := mapping function of a df $\in$ HDFIP to an appropriate pf $\in$ PF.	
α := mapping function of an af $\in$ HAFIP to an appropriate pf $\in$ PF.	
$PDO := \{(do, po) \in HDO \times PO: po = \mu(do)\}$	[(perceptible) represented DO]
$PAO := \{(ao, po) \in HAO \times PO: po = \nu(ao)\}$	[(perceptible) represented AO]
μ := mapping function of a dialog object do $\in$ DO to an appropriate po $\in$ PO.	
ν := mapping function of an application object ao $\in$ AO to an appropriate po $\in$ PO.	

An interactive system can be distinguished in a dialog and an application manager (Edmonds and Hagiwara, 1990). Belonging to this differentiation we distinguish between two types of objects and two types of functions: dialog object (DO, e.g. "window") and application object (AO, e.g. "text document"), and dialog function (DF, e.g. "open window") and application function (AF, e.g. "insert section mark"). Each function has a functional interaction point (FIP): AF –> AFIP, DF –> DFIP. PF is the set of all implemented representations of FIPs. The "interaction point (IAP)" introduced by Denert (1977) is not differentiated enough to appropriately describe graphical user interfaces; an IAP is more or less the same as the "actual dialog context (DC)" discussed in this paper (see Figure 3, Figure 4, Figure 5, Figure 6, and Figure 8).

A perceptible AFIP is called a PAFIP and a perceptible DFIP is called a PDFIP (see Table 1). These perceptible structures can have visible, audible and/or tactile representations. PO is the set of all implemented representations of DOs (e.g. "button", "icon", "window", etc.) and AOs (e.g. "text document", "graphic", "data base", etc.). A perceptible AO is called a PAO and a perceptible DO is called a PDO. An AFIP changes the state of an AO, and a DFIP changes the state of a DO. All DFIPs are more or less "interactive overhead". DFIPs are only suitable to handle one of the most constrained interactive resource, namely the *screen space*. The complete set of all description terms is defined in Table 1 (for a more detailed version see Rauterberg, 1995a).

If both mapping function's δ and α are of the type 1:m(any), then the user interface is a command interface (see Figure 4) where the command interface has only one pf$\in$PF, the "command prompt" (e.g. the PF in Figure 3). If both mapping function's δ and α are of the type 1:1, then the user interface is a menu or direct manipulative interface where each f$\in$FS is related to a perceptible structure PF on the I/O-interface (see Figue 0). One important difference between a menu and a direct manipulative interface is the "interactive directness". A user interface is 100% interactively direct, if the user has fully access in the actual dialog context to all AFIPs (Laverson, Norman and Shneiderman, 1987). Good interface design is characterized by optimising the multitude of DFIPs (e.g. "flatten" the menu tree; Paap and Roske-Hofstrand, 1988) and by allocating an appropriate PDFIP to the remaining HDFIPs.

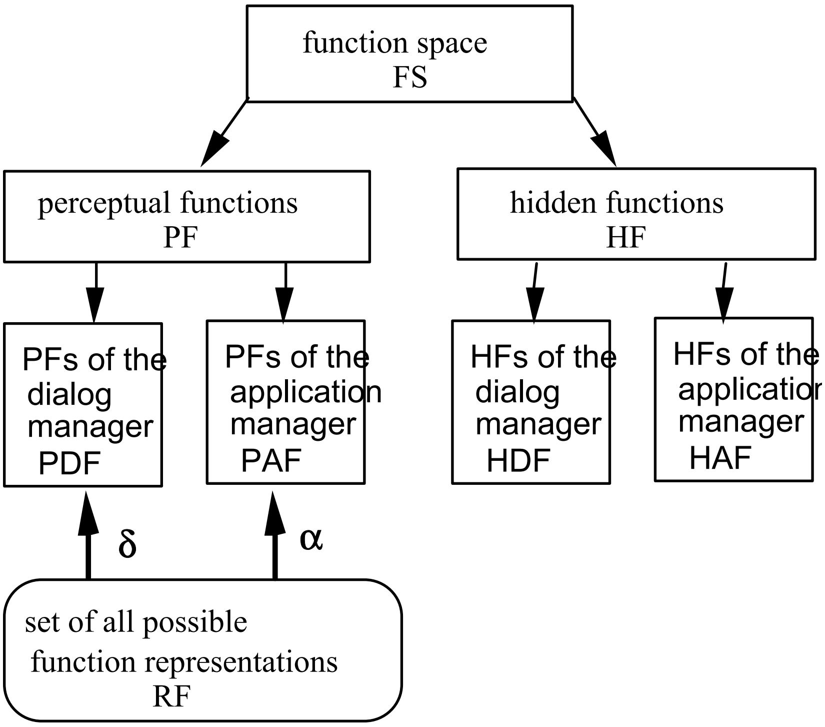

function space
FS
perceptual functions
PF
hidden functions
HF
PFs of the dialog manager PDF
PFs of the application manager PAF
HFs of the dialog manager HDF
HFs of the application manager HAF
δ
α
set of all possible function representations
RF

Figure 2. An overview over the four differtent sets of interaction points of the function space.

In the context of an actual dialog state the user must know what he or she can do next. To support the user in this way, different kinds of representational structures for functions (PF, e.g. "menus", "icons") have been developed (see Figure 5). If each functional interaction point (FIP) has its own representational interaction point (PF), then the user has 100% feedback (fFB) of all available functions. To estimate the amount of "feedback" of an interface a ratio is calculated: "number of PFs" (#PF = #PDFIP + #PAFIP) divided by the "number of HFs" (#HF = #HDFIP + #HAFIP) per dialog context. This ratio quantifies the average "amount of feedback" of the function space (fFB). (D is the number of all different dialog contexts.)

$$\mathrm{fFB} = 1/D \mathop{\bullet}\limits_{d=1}^{D} (\#PF_d / \#HF_d) * 100\% \qquad \textbf{[(functional) feedback]}$$

The physical limitation of the I/O-interface (screen size) is one reason, not to present all available functional interaction points (FIPs) with a specific representation (PF) on the screen. So, the user has to navigate through menu structures (= activating DFIPs) to come down to a DC with the desired AFIP (cf. Figure 5). The average length (lng) of "nearly" all possible sequences of dialog operations (PATH) from the top level dialog context down to DCs with the desired AFIP can be used as a good quantitative metric of "interactive directness" (ID): the reciprocal value of the average path length (lng = number of dialog steps). "Nearly" means that not all possible paths are included in this calculation, but only really used paths. An interface with the maximum ID of 100% has only one DC with path lengths of 1 dialog step. (P is the number of all different dialog PATHs.)

$$\mathrm{ID} = \left\{ 1/P \mathop{\bullet}\limits_{p=1}^{P} \mathrm{lng}(\mathrm{PATHp}) \right\}^{-1} * 100\% \qquad \textbf{[interactive directness]}$$

The number of ways to leave a DC is a precise measure of "dialog flexibility" ("fan" degree). To quantify the flexibility of the dialog manager we calculate the average number of DFIPs per dialog context (DFD). [D = number of all different dialog contexts]:

```
             D
DFD = 1/D    •   (#DFIPd)            [flexibility of the dialog manager]
             d=1
```

To quantify the flexibility of the application interface we calculate the average number of AFIPs per dialog context (DFA). A modeless dialog state has maximal dialog flexibility (e.g., "command" interfaces, or Oberon [7]). [D = number of all different dialog contexts]:

```
             D
DFA = 1/D    •   (#AFIPd)        [flexibility of the application manager]
             d=1
```

We carried out three different comparative usability studies to validate our measures. A fourth external comparative study was used for cross validation of our measures (see Rauterberg, 1995a and 1995b). All four investigated software products were given with the same application kernel, but two different interfaces each (the dialog managers, resp.). Given the characteristic values for feedback and flexibility of both user interfaces per application kernel, we are able to predict the outcome of a comparison study based on a performance metric (e.g., "task solving time").

Today several dialog techniques are developed and in usage. The following dialog techniques and dialog objects can be distinguished with regard to traditional user interfaces: command language, function key, menu selection, icon, and window (Shneiderman, 1987). These techniques can be summarised into three different *interaction styles:*

3 A Classification Schema for User Interfaces

Using the two quantitative measures "functional feedback" and "interactive directness" it is possible to classify the most common interface types: batch, command, menu, desktop (see Table 2). The command language interface is characterized by high interactive directness, but this interface type has a very low amount of visual feedback. Only graphical interfaces (GUIs) can support the user with sufficient visual feedback and with high interactive directness, too (c.f. Rauterberg, 1993 and Ulich, Rauterberg, Moll, Greutmann and Strohm, 1991).

Table 2. A classification schema of most common user interfaces.

[visual] feedback (FB)

		low	**high**
interactive directness (ID)	**low**	batch	menu interface MI
			desktop style
	high	command language CI	direct manipulation DI

To make this classification schema as understandable as possible, we describe the three classified interfaces (1) with one representative example of a concrete product and (2) with an abstract schema of the dialog structure.

If this classification schema is valid, then we should observe the following outcomes of empirical comparison studies: a command language interface--with a maximum of interactive directness--should not always outperfom a menu interface; sometimes should a menu interface--with a maximum of functional feedback--be superior to a command interface. The outcomes of several comparison studies (CI versus MI) should be heterogenous.

That a desktop interface (in general: a graphical user interface, a GUI) has a higher usability performance than a menu interface (in general: a character based user interface, a CUI) could be shown in Rauterberg (1992). Today, it seems to be common sense, that GUIs have a better usability performance then CUIs. But, that GUIs should also be better then command interfaces, is one of the open question. Especially experts deny this statement!

If the desktop like interfaces--with high functional feedback and high interactive directness--are really better than command language interfaces, then we should find most of the outcomes of empirical comparison studies in this direction!

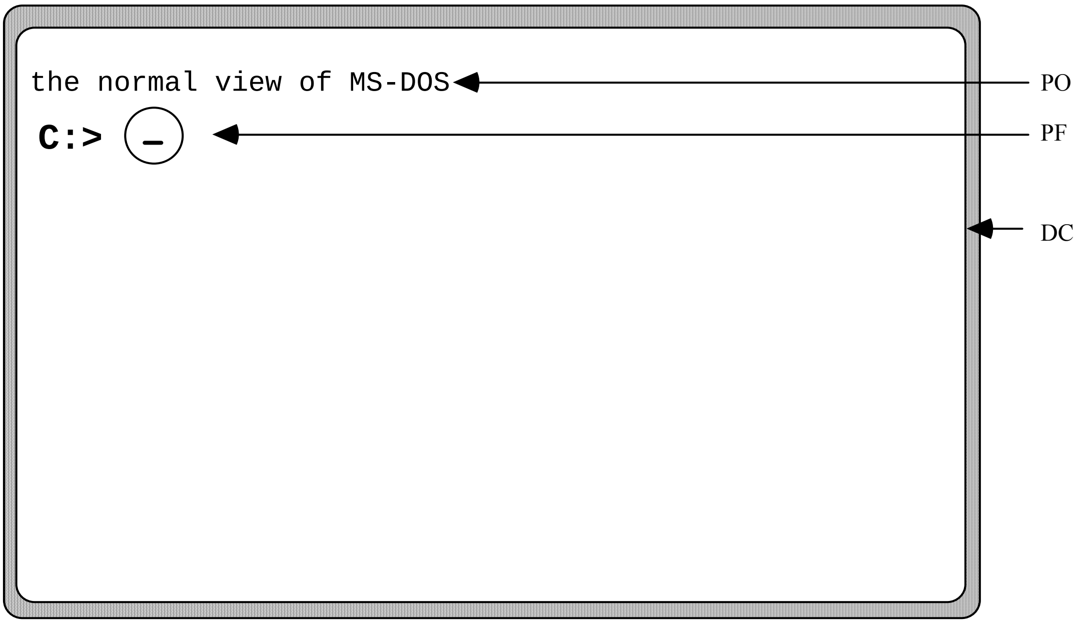
4 Description of Different User Interfaces

4.1 Command [language] interfaces (CI)

This interaction style by typing in words from a set of legal commands is one of the oldest way to interact with a computer. If some or all the options and function points of a menu interface may be accessed directly through keyboard equivalents (including action codes, function keys, and softkeys) then we call this interface also a command-like interface.

Pros: In the command mode the user has a maximum of *direct access* to all available functions and operations. This directness can be measured with the metric ID ≈ 1 (see Figure 4).

Cons: The user has no permanent feedback of all actual available function points This aspect can be measured with the metric fFB « 1 (for example see Figure 3).



```
the normal view of MS-DOS
C:> _
```

Figure 3. An actual dialog context (DC) of the operating system MsDOS with the representation space of the interactive object (PO = PDO ∪ PAO: "text output") and the

representation space of the interactive functions (PF = PDFIP ∪ PAFIP: "command entry point" marked by a circle).

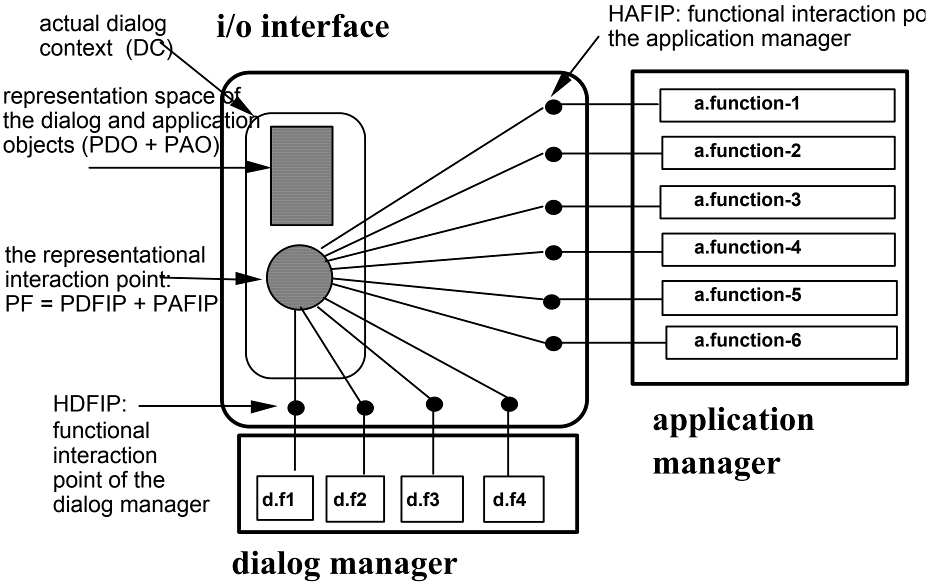

actual dialog context (DC)
i/o interface
HAFIP: functional interaction pc the application manager
representation space of the dialog and application objects (PDO + PAO)
a.function-1
a.function-2
a.function-3
a.function-4
a.function-5
a.function-6
the representational interaction point: PF = PDFIP + PAFIP
HDFIP: functional interaction point of the dialog manager
application manager
d.f1
d.f2
d.f3
d.f4
dialog manager

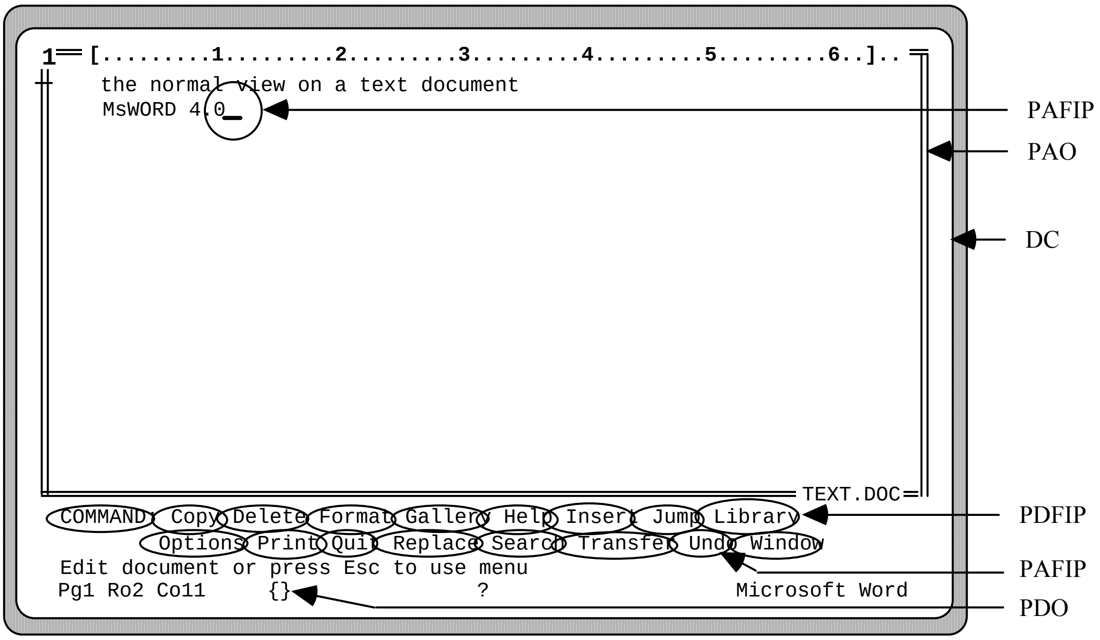
Figure 4. An idealised schema of a command interface with a fictive I/O-interface, a dialog and an application manager.

4.2 Menu interface (MI)

This interaction style includes rigid menu structures, pop-up and pull-down menus, form fill-in, etc. This style became technically possible only with those terminals that, essentially, can reproduce only the ASCII character set. With this type of interaction style function keys are often used in addition to manage the dialog.

Pros: Most available functions are represented by perceivable interaction points (PF's). This feature can be measured with the metric fFB ≈ 100% (see Figure 5).

Cons: Finding a function point in deeper menu hierarchies is cumbersome; this can be measured with the metric ID » 1 (for example see Figure 6).



```
1══[.........1.........2.........3.........4.........5.........6..]..
    the normal view on a text document
    MsWORD 4.0_
                                                      TEXT.DOC
 Edit document or press Esc to use menu
 Pg1 Ro2 Co11     {}             ?                 Microsoft Word
```

Figure 5. An actual dialog context (DC) of the text processing program MS-Word with the representation space of the interactive object (PAO: "text document"; PDO: "clipboard"), and the representation space (PF: marked by circles) of the interactive functions (PAFIP: "text entry point", "undo"; PDFIP: menu options).

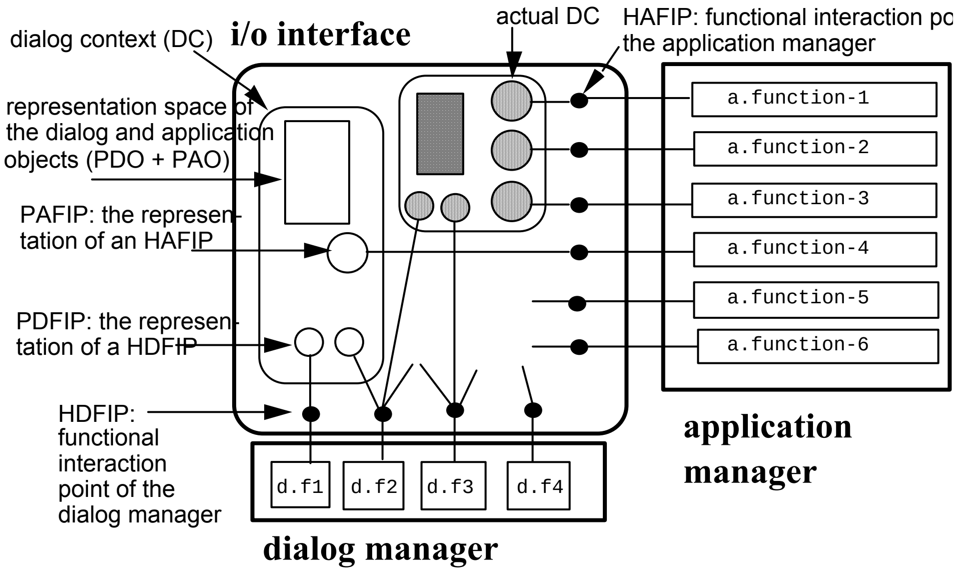


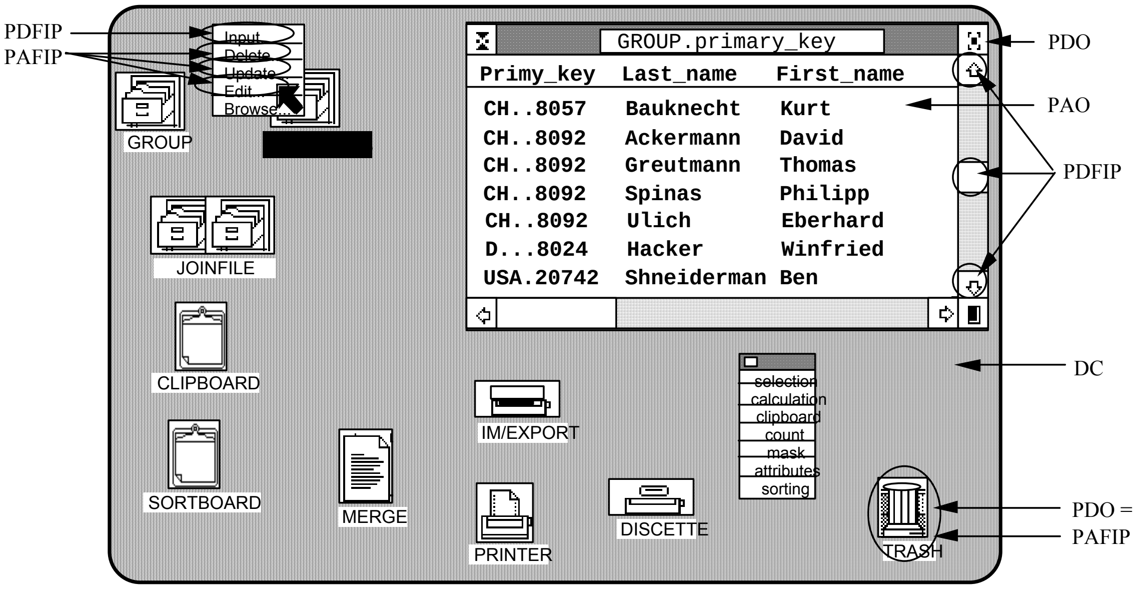
Figure 6. An idealised schema of a menu interface with a fictive I/O-interface, a dialog and an application manager.

4.3 Direct manipulative interface (DI)

The development of this interaction style was based on the desktop metaphor which assumes that by depicting the work environment (i.e. of the desk: files, waste-paper basket, etc.) as realistically as possible on the I/O-interface, it would be particularly easy for the user to adjust to the virtual world of electronic objects.

Pros: All functions are represented by visible interaction points. The activation of intended functions can be achieved by directly pointing to their visible representations (see Figure 7).

Cons: Direct manipulation interfaces have difficulty handling variables, or distinguishing the depiction of an individual element from a representation of a set or class of elements.



GROUP.primary_key

Primy_key	Last_name	First_name
CH..8057	Bauknecht	Kurt
CH..8092	Ackermann	David
CH..8092	Greutmann	Thomas
CH..8092	Spinas	Philipp
CH..8092	Ulich	Eberhard
D...8024	Hacker	Winfried
USA.20742	Shneiderman	Ben

Figure 7. An actual dialog context (DC) of a direct manipulative interface with the representation space of the interactive object (PAO: e.g., data window; PDO: e.g., trash), and the representation space (PF: marked by circles) of the interactive functions (PAFIP: e.g., pop-up menu, trash; PDFIP: e.g., window scrolling).

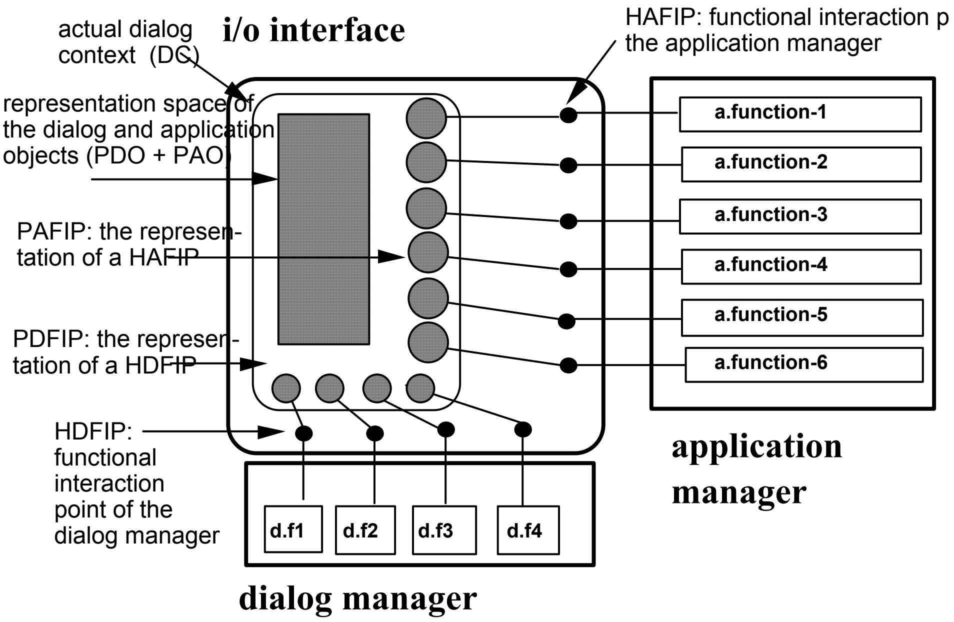


Figure 8. An idealised schema of a direct manipulative interface with a fictive I/O-interface, a dialog and an application manager.

5 Empirical Validation of the Classification Schema

A major task in our area of HCI is the development of a theoretical explanation of the outcomes presented in Table 3 and Table 4 where we have available the results of a number of previous studies. Our first task is to find out what empirical relationships have been revealed in these studies so we can take them into account. In developing an understanding of these relationships, it is helpful in reviewing the studies to make up a table summarising the findings. Table 3 and Table 4 show such summaries. In addition to the observed empirical outcome we recorded data on (1) compared interaction styles, (2) skill levels, (3) performance or attitude metrics, (4) the direction of the outcome, and (5) the result of the statistical test.

First, we present an overview of the results of eight different empirical investigations which compared a command (CI) with a menu (MI) interface (see Table 3). To measure differences in the usage and in the personnel opinion several different metrics are used: task solving time, error rate, number of slips, error correction time, and subjective rating (for further details see in the references).

Table 3. The outcomes of nine different comparison studies between command (CI) and menu (MI) interfaces. ("CI > MI" means that the average usage/preference with/for CI is better than with/for MI; "CI < MI" means that the average usage/preference with/for MI is better than with/for CI; "CI = MI" means that there are no published data to decide; "sig." means that p ≤ 0.05; "not sig." means that p > 0.05)

Reference	**interface**	**skill level**	**usability metric**	**outcome**	**test result**
Streitz et al. (1987)	CI, MI	beginner	task solving time	CI < MI	sig.
Chin et al. (1988)	CI, MI	beginner	subjective rating	CI < MI	sig.
Ogden and Boyle (1982)	CI, MI, HY	beginner	preferences	CI < MI	sig.
Roy (1992)	CI, MI	advanced	error rate	CI < MI	sig.
Roberts and Moran (1983)	CI, MI, DI	experts	task solving time	CI < MI	sig.
Chin et al. (1988)	CI, MI	experts	subjective rating	CI < MI	sig.
Peters et al. (1990)	CI, MI, DI	experts	slips	CI < MI	sig.
Peters et al. (1990)	CI, MI, DI	experts	recognition errors	CI < MI	sig.
Peters et al. (1990)	CI, MI, DI	experts	efficiency	CI < MI	sig.
Ogden and Boyle (1982)	CI, MI, HY	beginner	task time	CI < MI	not sig.
Roy (1992)	CI, MI	advanced	task solving time	CI < MI	not sig.
Antin (1988)	CI, MI, KMI	advanced	subjective rating	CI < MI	not sig.
Hauptmann & Green (1983)	CI, MI, NO	beginner	task solving time	CI = MI	not sig.
Hauptmann & Green (1983)	CI, MI, NO	beginner	number of errors	CI = MI	not sig.
Hauptmann & Green (1983)	CI, MI, NO	beginner	subjective rating	CI = MI	not sig.
Whiteside et al. (1985)	CI, MI, IO	beginner	task completion rate	CI > MI	not sig.
Antin (1988)	CI, MI, KMI	advanced	preferences	CI > MI	not sig.
Roberts and Moran (1983)	CI, MI, DI	experts	error-free task time	CI > MI	not sig.
Whiteside et al. (1985)	CI, MI, IO	advanced	task completion rate	CI > MI	sig.
Streitz et al. (1987)	CI, MI	advanced	task solving time	CI > MI	sig.
Antin (1988)	CI, MI, KMI	advanced	task completion rate	CI > MI	sig.
Whiteside et al. (1985)	CI, MI, IO	experts	task completion rate	CI > MI	sig.

The general result of this first overview (Table 3) is that there is no clear advantage neither for CI nor for MI. In nine of twenty-two measurements (41%) we can observe a clear advantage for MI, and in nine of twenty-two measurements (41%) are no significant differences; but, in four of twenty-two measurements (18%) there are significant advantages for CI.

Second, we present an overview of the results of twelve different empirical investigations which compared a command (CI) with a direct manipulative (DI) interface (see Table 4). To measure differences in the usage and in the personnel opinion several different metrics are used: task solving time, number of errors, time between errors, error correction time, efficiency, and subjective rating (for further details see in the references).

The general result of this second overview (Table 4) is that DI seems to be generally better than CI, not only for beginners, but also for advanced and expert users. In nineteen of twenty-five measurements (76%) we can observe an advantage for DI; in five of twenty-five measurements (20%) are no significant differences; and, only in one measurement (4%) is a significant advantage for CI.

Table 4. The outcomes of twelve different comparison studies between command (CI) and desktop and direct manipulative (DI) interfaces. ("CI > DI" means that the average usage/preference with/for CI is better than with/for DI; "CI < DI" means that the average usage/preference with/for DI is better than with/for CI; "CI = DI" means that there are no published data to decide; "sig." means that p ≤ 0.05; "not sig." means that p > 0.05)

Reference	interface	skill level	usability metric	outcome	result
Altmann (1987)	CI, DI	beginner	task solving time	CI < DI	sig.
Karat et al. (1987)	CI, DI	beginner	task solving time	CI < DI	sig.
Streitz et al. (1989)	CI, DI	beginner	task solving time	CI < DI	sig.
Sengupta & Te'eni (1991)	CI, DI	beginner	task solving time	CI < DI	sig.
Margono et al. (1987)	CI, DI	beginner	number of errors	CI < DI	sig.
Morgan et al. (1991)	CI, DI	beginner	number of errors	CI < DI	sig.
Morgan et al. (1991)	CI, DI	beginner	time between errors	CI < DI	sig.
Karat et al. (1987)	CI, DI	beginner	error correction time	CI < DI	sig.
Morgan et al. (1991)	CI, DI	beginner	error-free time	CI < DI	sig.
Margono et al. (1987)	CI, DI	beginner	subjective rating	CI < DI	sig.
Morgan et al. (1991)	CI, DI	beginner	subjective rating	CI < DI	sig.
Torres-Chazaro et al.(1992)	CI, DI	beginner	subjective rating	CI < DI	sig.
Sengupta & Te'eni (1991)	CI, DI	beginner	efficient usage	CI < DI	sig.
Tombaugh et al. (1989)	CI, DI	advanced	subjective rating	CI < DI	sig.
Torres-Chazaro et al.(1992)	CI, DI	advanced	subjective rating	CI < DI	sig.
Roberts and Moran (1983)	CI, MI, DI	experts	task solving time	CI < DI	sig.
Peters et al. (1990)	CI, MI, DI	experts	oblivion's errors	CI < DI	sig.
Peters et al. (1990)	CI, MI, DI	experts	recognition error	CI < DI	sig.
Peters et al. (1990)	CI, MI, DI	experts	efficiency	CI < DI	sig.
Margono et al. (1987)	CI, DI	beginner	task solving time	CI < DI	not sig.
Morgan et al. (1991)	CI, DI	beginner	task solving time	CI < DI	not sig.
Tombaugh et al. (1989)	CI, DI	advanced	task solving time	CI < DI	not sig.
Roberts and Moran (1983)	CI, MI, DI	experts	error correction time	CI < DI	not sig.
Altmann (1987)	CI, DI	beginner	subjective rating	CI > DI	not sig.
Masson et al. (1988)	CI, DI	advanced	task solving time	CI > DI	sig.

6 Discussion

To come to a conclusion which interface style is the best, we need a lot of empirical studies. But, the most empirical studies have one of the following weaknesses (Paap and Roske-Hofstrand, 1988, p.207): Two or more commercially available systems are compared, which have different application managers (e.g., Whiteside et al., 1985; Altmann, 1987), or two or more different interfaces of the same application manager are evaluated, but these systems are only prototypes in a laboratory setting (e.g., Streitz, Spijkers and van Duren, 1987). Another problem seems to be the selection of real expert users. So normally empirical investigations are done with beginners only (e.g., Margono and Shneiderman, 1987; Streitz, Lieser and Wolters, 1989), and if the investigation tries to explain the differences between beginners and experts, trained beginners are mostly declared as experts. So, we classified "trained beginners" as "advanced" users, and the term "experts" was reserved only for users with long personal experiences in using the investigated systems.

Since so far sufficient results are available with respect to a comparison of user interfaces based (1) on command interfaces, (2) on conventional menu selection, and (3) on direct manipulative interfaces, these three interaction styles were compared in this paper. To test the often expressed opinion, that desktop interfaces are only good for beginners--and not for experts--(Hutchins, Hollan and Norman, 1986, p. 117), this aspect should be considered, too.

If the classification of the three most common interfaces in chapter 3 is valid (see above), then we expect different outcomes of empirical comparison studies. On the side of interactive directness, the command interface is superior to menu interfaces; on the other side of functional feedback, the menu interface must show significant advantages. It is impossible to compare both interfaces empirically by separating the two factors--functional feedback and interactive directness--without destroying the characteristic of each interface style. This overlay of the two independent factors may be one reason for the--seeming--incongruent and inconsistent results in Table 3.

One of the main goal of research in this area is the production of an integrated statement of the empirical findings of the many pieces of research done. In a broad sense, this means a theoretical analysis of how and why the many facts fit together. However, our quantitative description based on interaction points--as a broad theoretical integration--cannot be put on a sound footing until a narrower integration of the cited empirical studies has taken place. This narrow focus on single empirical outcomes of several comparison studies is the starting point for a *meta-analysis* (Rosenthal, 1984).

Table 5: Contingency table of a meta-analysis only for significant differences (result = "sig.").
[CELL CONTENT: observed frequency (expected frequency)]

	MI	**DI**	**outcome of this meta-analysis**
CI better as ...	4 (2.0)	1 (3.0)	Chi** = 4.07, df = 1
CI worse as ...	9 (11.0)	19 (17.0)	p ≤ .044

To estimate the correlation between (1) the type of the comparison ("CI versus MI" or "CI versus DI") and (2) the direction of the outcome ("CI better as MI or DI" versus "CI worse as MI or DI"), we calculated the Chi-square test of the appropriate contingency table (Table 5). We can find a significant correlation between both dimensions (p ≤ .044; see Table 5). This correlation means that CI has a higher chance to be better if it is compared with MI, and--on the other side--a significant lower chance to outperform DI. This meta-analytical result is a strong evidence that our classification schema (see Table 2) is one possible, plausible, and consistent interpretation. Therefore, we interpret this result as an *empirical validation* of our two metrics fFB and ID.

Table 6: Contingency table of a meta-analysis only for significant differences (result = "sig.").
[CELL CONTENT: observed frequency (expected frequency)]

	beginner	advanced+	outcome of this meta-analysis
CI better as MI/DI	0 (2.4)	5 (2.6)	Chi** = 5.55, df = 1
CI worse as MI/DI	16 (13.6)	12 (14.4)	$p \leq .018$

To find out which interaction style is appropriate for which skill level of the user, we analysed the contingency table with the two dimensions: (1) direction of the outcome ("CI better as [MI or DI]" versus "CI worse as [MI or DI]"), and (2) skill level of the users ("beginner" versus "advanced + experts" = "advanced+"). We can find a significant correlation between both dimensions ($p \leq .018$; see Table 6). This correlation means that the outcome "CI better as [MI or DI]" can be significantly more often observed with advanced test users than with beginners. This result is a first and strong empirical confirmation of the often expressed opinion that CI is especially good for experts.

7 Conclusion

Standards and norms need product oriented operationalization of interface features. To attain this goal, a description language for interface structures which is general enough to classify the different interface types and detailed enough to allow quantification is required. The descriptive concept for functional "interaction points" (FIP), which is introduced in this paper, meets these both conditions. The function space (FS) is a set of all implemented FIPs and can be distinguished in (1) functional and representational interaction points, and (2) dialog and application specific interaction points. The degree of visualisation and interactive directness can be described and measured based on these interaction points.

Using the two quantitative metrics "functional feedback" (fFB) and "interactive directness" (ID) in measuring two relevant aspects of user interactive quality it is possible to classify the most common interface types: [batch], command, menu, desktop. The command interface is characterized by high interactive directness, but has a very low amount of functional feedback. Only graphical interfaces (GUIs) can support the user with sufficient interactive directness and with high visibility.

In addition to the metrics for "functional feedback" and "interactive directness" two other quantitative metrics have been defined and validated: "flexibility of the dialog interface" and "flexibility of the application interface" (Rauterberg, 1993). The empirical validation of these two additional measures was carried out with six different I/O-interfaces of six different dialog managers for three different application managers ("relational data base system", "multi media information system", and "simulation tool kit"; detailed description in Rauterberg, 1995a and 1995b).

The presented approach to quantify usability attributes and the interactive quality of user interfaces in a task independent way is a first step in the right direction. The next step is a more detailed analysis of the relevant characteristics and validation of these characteristics in further empirical investigations. Standardised criteria need to be developed to test user interfaces for conformity with standards.

8 References

Altmann, A. (1987) Direkte Manipulation: empirische Befunde zum Einfluss der Benutzeroberfläche auf die Erlernbarkeit von Textsystemen. Zeitschrift für Arbeits- und Organisationspsychologie 31(3):108-114.

Antin, J. (1988) An empirical comparison of menu selection, command entry and combined modes of computer control. Behaviour and Information Technology 7(2):173-182.

Bevan, N., Kirakowski, J. & Maissel, J (1991) What is Usability? in: Human Aspects in Computing: Design and Use of Interactive Systems with Terminals; (Bullinger, H-J.; ed.); Amsterdam: Elsevier; 651-655.

Chin, J. P., Diehl, V. A. & Norman, D. (1988) Development of an instrument measuring user satisfaction of the human-computer interface. In: E. Soloway, D. Frye & S. B. Sheppard (eds.) Human Factors in Computing Systems - CHI'88. New York: ACM,, pp. 213-218.

Denert, E. (1977) Specification and design of dialogue systems with state diagrams. in: International Computing Symposium 1977; (Morlet, E. & Ribbens, D.; eds.); Amsterdam: North-Holland; 417-424.

Edmonds, E. & Hagiwara, N. (1990) An experiment in interactive architectures. In: Human-Computer Interaction - INTERACT '90. (Diaper, D. et al.; eds.) Amsterdam: Elsevier; 601-606.

Hauptmann, A. G. & Green, B. F. (1983) A comparison of command, menu-selection and natural-language computer programs. Behaviour and Information Technology 2(2): 163-178.

Jeffries, R. & Desurvire, H. (1992) Usability testing vs. heuristic evaluation: was there a contest? SIGCHI Bulletin 24(4), 39-41.

Karat, J. (1988) Software Evaluation Methodologies. in: Handbook of Human-Computer Interaction; (Helander, M.; ed.); Amsterdam: Elsevier; 891-903.

Karat, J., Fowler, R. & Gravelle, M. (1987) Evaluating user interface complexity. In: In: H-J. Bullinger & B. Shackel (eds.) Human-Computer Interaction - INTERACT '87. Amsterdam: North-Holland, pp. 489-495.

Kirakowski, J. & Corbett, M. (1990) Effective Methodology for the Study of HCI. in: Human Factors in Information Technology, vol. 5; (Bullinger, H. & Polson, P.; eds.); Amsterdam: North-Holland.

Kirakowski, J. and Corbett, M. (1993) SUMI: The Software Usability Measurement Inventory, British Journal of Educational Technology, 24(3), 210-212.

Laverson, A., Norman, K. & Shneiderman, B. (1987) An evaluation of jump-ahead technique in menu selection. Behaviour and Information Technology 6(2), 97-108.

Margono, S. & Shneiderman, B. (1987) A study of file manipulation by novices using commands vs. direct manipulation. In: Proceedings of 26th Annual Technical Symposium, Washington D.C. Chapter of ACM Gaithersburg, MD - June, 11, 1987. New York: ACM, pp. 154-159.

Morgan, K., Morris, R. & Gibbs, S. (1991) When does a mouse become a rat? or … comparing performance and preferences in direct manipulation and command line environment. The Computer Journal 34(3):265-271.

Masson, M., Hill, W., Conner, J. & Guidon, R. (1988) Misconceived misconception? In: E. Soloway, D. Frye & S. Sheppard (eds.) Human Factors in Computing Systems - CHI'88. New York: ACM, pp. 151-156.

Nielsen, J & Mack, R. (1994, eds.) Usability inspection methods.New York: Wiley.

Ogden, W. & Boyle, J. (1982) Evaluating human-computer dialog styles: command vs. form/fill-in for report modification. In Proceedings of the 26the Annual Meeting of the Human Factors Society, pp. 542-545.

Paap, K. & Roske-Hofstrand, R. (1988) Design of menus. in: Handbook of Human-Computer Interaction; (Helander, M.; ed.); Amsterdam: North-Holland; 205-235.

Peters, H., Frese, M. & Zapf, D. (1990) Funktions- und Nutzungsprobleme bei unterschiedlichen Dialogformen. Zeitschrift für Arbeitswissenschaft 44(3):145-152.

Rauterberg, M. (1992) An empirical comparison of menu-selection (CUI) and desktop (GUI) computer programs carried out by beginners and experts. Behaviour and Information Technology 11(4), 227-236.

Rauterberg, M. (1993) A product oriented approach to quantify usability attributes and the interactive quality of user interfaces. In: H. Luczak, A. Cakir & G. Cakir (Eds.) Work With Display Units 92. Amsterdam: North-Holland, pp. 324-328.

Rauterberg, M. (1995a) Four different measures to quantify three usability attributes: 'feedback', 'interactive directness' and 'flexibility'. In: P. Palanque & R. Bastide (eds.) Design Specification and Verification of Interactive Systems'95. Wien New York: Springer, pp. 209-223.

Rauterberg, M. (1995b) Ein Konzept zur Quantifizierung software-ergonomischer Richtlinien. Zürich: IfAP ETH-Press.

Rengger, R. (1991) Indicators of usability based on performance. in: Human Aspects in Computing: Design and Use of Interactive Systems with Terminals; (Bullinger, H-J.; ed.); Amsterdam: Elsevier; 656-660.

Roberts, T. & Moran, T. (1983) The Evaluation of Text Editors: Methodology and Empirical Results. Communications of the ACM 26(4):265-283.

Rosenthal, R. (1984) Meta-analysis procedures for social research. (Applied Social Research Methods Series, Vol. 6), London: Sage.

Roy, G. (1992) An evaluation of command line and menu interface in a CAD environment. International Journal of Computer Integrated Manufacturing 5(2):94-106.

Sengupta, K. & Te'eni D. (1991) Direct manipulation and command language interfaces: a comparison of users' mental models. In: H.-J. Bullinger (ed.) Human Aspects in Computing: Design and Use for Interactive Systems and Information Management. (Advances in Human Factors/Ergonomics, 18A, pp. 429-434), Amsterdam : Elsevier.

Shneiderman B (1987) Designing the User Interface. Addison-Wesley, Reading MA.

Streitz, N., Spijkers, W. A. C. & van Duren, L. L. (1987) From Novice to expert user: a transfer of learning on different interaction modes. In: H.-J. Bullinger & B. Shackel (eds.) Human-Computer Interaction - INTERACT ’87. New York: North-Holland, pp. 841-846.

Streitz, N., Lieser, A. & Wolters, A. (1989) The combined effects of metaphor worlds and dialogue modes in human-computer-interaction. In: F. Klix, N. Streitz, Y. Waern & H. Wandke (eds.) Man-Computer Interaction Research MACINTER-II. Amsterdam: Elsevier, pp. 75-88.

TCO (1992) Software Checker, vers. 2.0. The Swedish Confederation of Professional Employees, Office address: Linnegatan 14, P.O. box 5252, S-102 45 Stockholm.

Tombaugh, J., Paynter, B. & Dillon, R. (1989) Command and graphic interfaces: user performance and satisfaction. In: G. Salvendy & M. Smith (eds.) Designing and Using Human-Computer Interfaces and Knowledge Based Systems. Amsterdam: Elsevier, pp. 369-375.

Torres-Chazaro, O., Beaton, R. & Deisenroth, M. (1992) Comparison of command language and direct manipulation interfaces for CNC milling machines. International Journal of Computer Integrated Manufacturing 5(2):107-117.

Ulich, E., Rauterberg, M., Moll, T., Greutmann, T. & Strohm, O. (1991): Task orientation and User-Oriented Dialog Design. International Journal of Human-Computer Interaction 3(2), 117-144.

Whiteside, J., Jones, S., Levy, P. S. & Wixon, D. (1985) User Performance with Command, Menu, and Iconic Interfaces. Human Factors in Computing Systems-II. Proceedings of the CHI ´85 Conference in San Francisco, Amsterdam New York Oxford: North-Holland, pp. 185-191.