WORK WITH DISPLAY UNITS 92
H. Luczak, A. Cakir and G. Cakir (Editors)



# A product oriented approach to quantify usability attributes and the interactive quality of user interfaces.

Matthias Rauterberg
Work and Organizational Psychology Unit, Swiss Federal Institute of Technology (ETH)
Nelkenstr. 11, CH-8092 Zurich, Switzerland

**Abstract**
Four different measures based on a product oriented view of user interfaces are introduced. Different types of user interfaces can be described and differentiated by the powerful concept of "interaction points". Regarding to the interactive semantic of "interaction points" (FIPs), 4 different types of FIPs must be discriminated. Based on the concept of interaction points, the dimensions "feedback", "interactive directness" and "flexibility" can be quantified. The results of a theoretical and an empirical validation of 4 quantitative measures are presented and discussed.

## 1. INTRODUCTION

One of the main problems of standards (e.g. ISO) in the context of software ergonomics is, that they can not be measured in a quantitative way. The scope of our paper encloses the product "interactive software". Today there are 4 different views on human computer interaction to measure interactive qualities (see also [10]; [1]:651).

(1) The user performance view: usability can be measured by examining how the user interacts with the product; all kinds of usability testings with "real" users are subsumed here [5].
(2) The *user oriented view*: usability can be measured in terms of the mental effort and attitude of the user (–> questionnaires and interviews).
(3) The *product oriented view*: usability can be measured in terms of the ergonomic attributes of the product (–> quantitative measures); all heuristic evaluations carried out by ergonomic experts investigating a concrete software product fall in this category, too.
(4) The *formal view*: usability can be formalized and simulated in terms of mental models; Karat (1988) describes formal methods in the context of "theory based" evaluation.

These 4 different views can be classified by the two dimensions "user" (u) and "computer" (c), which are really (r) or virtually (v) involved in the measurement: (1) = u.r & c.r, (2) = u.r & c.v, (3) = u.v & c.r, and (4) = u.v & c.v. Till today nobody knows, how these 4 different views can be successfully combined (see [4]). Nevertheless, "there are invariable errors or misconceptions which will be discovered in any process of having users test a system that even the most insightful of designers would miss. It is a necessary experience to become aware of the fact that none of us knows everything about how the human mind functions" ([5]:895).

## 2. A DESCRIPTIVE CONCEPT OF INTERACTION POINTS

To come up with a set of metrics of usability based on the product oriented view, a concept of descriptive terms, which can be counted, is necessary. The granularity of the descriptive terms must be on a medium level: not to specific (e.g. "push button", "menu option", etc.), and not to general (e.g. "transparent", "flexible", etc.). We need a granularity level, on which we can describe the different types of user interfaces ("batch", "command", "menu", "desktop") in a uniform and precise way, which is powerful enough and easy to go.

The interaction space (IS) consists of two different interlaced spaces: the object space (OS), and the function space (FS). OS encloses all perceptible objects (RO) 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 functions (RF) and hidden functions (HF). A dialogue context (DC) contains a subset of the union of OS and FS.

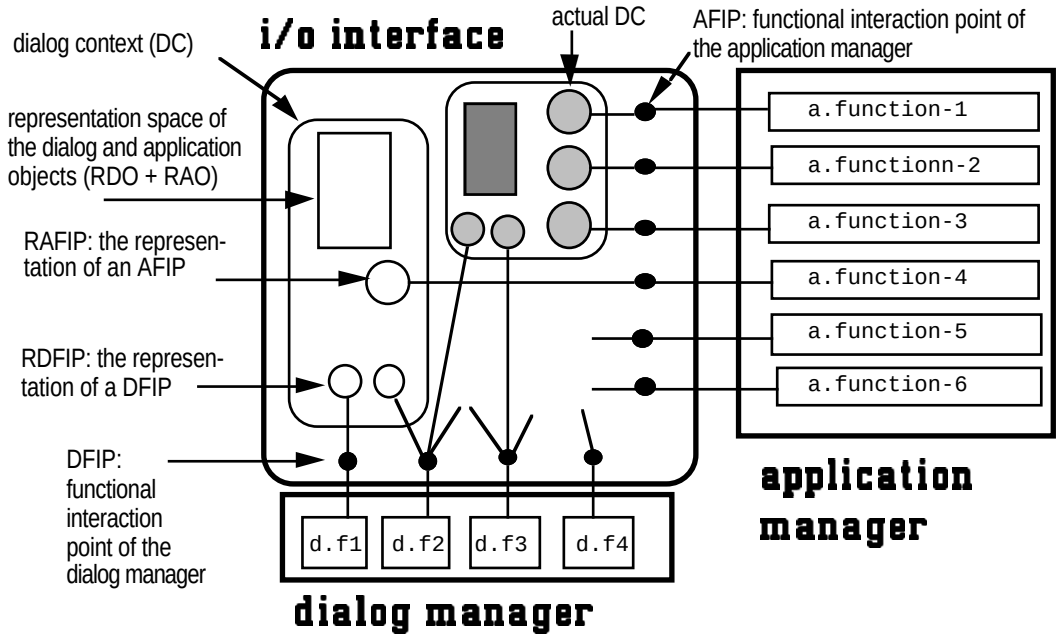


Figure 1. A schematic presentation of the i/o interface, the dialog and the application manager of an interactive system with a two level menu tree.

Belonging to the distinction of an interactive system in a dialog and an application manager [3], we distinguish between two types of objetcs and two types of functions: dialog object (DO) and application object (AO), and dialog function (DF) and application function (AF). Each function has a functional interaction point (FIP): AF –> AFIP, DF –> DFIP. RF is the set of all implemented representations of FIPs. A perceptible AFIP is called a RAFIP, a perceptible DFIP is called a RDFIP (see Figure 1 and 2). These perceptible structures can have a visible, audible and/or tactile representation. RO is the set of all implemented representations of DOs (e.g. button, icon, window, etc.) and AOs (e.g. text document, grafic, data base, etc.). A perceptible AO is called a RAO, a perceptible DO is called a RDO. 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 the constrained interactive ressources (e.g. screen space).

The complete set of all description terms is defined as follows:

$IS := OS \cup FS$; $DC \in IS$; $OS := RO \cup HO$; $FS := RF \cup HF$;

$RO := RDO \cup RAO$; $HO := DO \cup AO$; $RF := RDFIP \cup RAFIP$; $HF := DFIP \cup AFIP$;

$RDFIP := \{(df,rf) \in DFIP \times RF: rf = \delta(df)\}$; $RAFIP := \{(af,rf) \in AFIP \times RF: rf = \alpha(af)\}$

$\delta$ := mapping function of a $df \in DFIP$ to an appropriate $rf \in RF$.

$\alpha$ := mapping function of an $af \in AFIP$ to an appropriate $rf \in RF$.

$RDO := \{(do,ro) \in DO \times RO: ro = \mu(do)\}$; $RAO := \{(ao,ro) \in AO \times RO: ro = \nu(ao)\}$

$\mu$ := mapping function of a dialog object $do \in DO$ to an appropriate $ro \in RO$.

$\nu$ := mapping function of an application object $ao \in AO$ to an appropriate $ro \in RO$.

The intersection of RF and RO is sometimes not empty: $RF \cap RO \neq \emptyset$; in the context of graphical interfaces are icons elements of this intersection: e.g. RDFIP "copy" ≡ RDO "clipboard", RAFIP "delete" ≡ RAO "trash".

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 "dialog context (DC)" discussed in this paper (see Figure 1 and 2).

If both mapping functions $\delta$ and $\alpha$ are of the type 1:m(any), then the user interface is a command interface; the command interface has only one $rf \in RF$: the "command prompt" (e.g. the

topmost RAFIP in Figure 2). If both mapping functions δ and α are of the type 1:1, then the user interface is a menu or direct manipulative interface; each f∈FS FIP is related to a perceptible structure RF on the i/o interface. 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 [7]. Good interface design is characterized by optimizing the multitude of DFIPs (e.g. "flatten" the menu tree [8]) and by allocating an appropriate RDFIP to the remaining DFIPs.

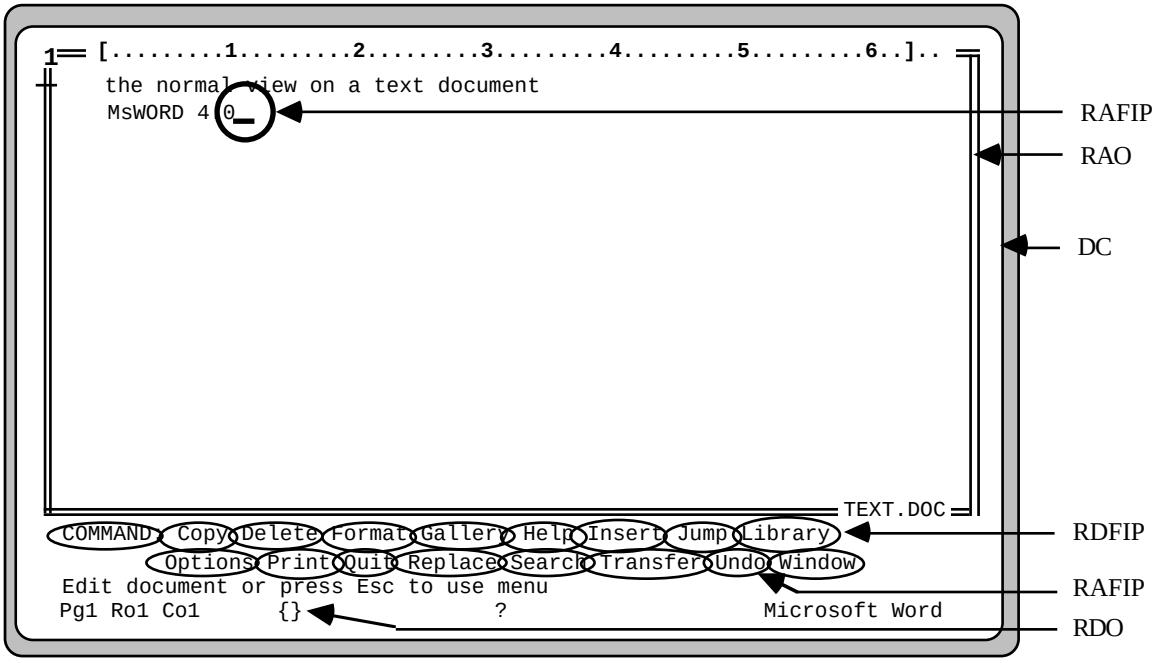


Figure 2. An actual dialog context (DC) of the textprocessing program MsWord with the representation space of the interactive object (RAO: text document; RDO: clipboard), and the representation space (RF: marked by circles) of the interactive functions (RAFIP: text entry point, undo; RDFIP: menu options).

### 2.1. A quantitative measure of "feedback" of the function space (FFB)

In the context of an actual dialog state the user must know, what he can do next. To support the user in this sense, different kinds of representational structures for functions (RF: e.g. menues, icons) have been developed. If each functional interaction point (FIP) has his own representational interaction point (RF), then the user has 100% feedback (FFB) of all available functions. To estimate the amount of "feedback" of an interface, we calculate a ratio: "number of RFs" (#RFIP = #RAFIP + #RDFIP) divided by the "number of FIPs" (#FIP = #AFIP + #DFIP) per dialog context (see Formula 1). This ratio quantifies the average "amount of feedback" of the function space (FFB). [D = number of all different dialog contexts]:

$$FFB = 1/D \sum_{d=1}^{D} (\#RFIP_d / \#FIP_d) * 100\% \qquad (1)$$

### 2.2. A quantitative measure of "interactive directness" (ID)

The physical limitations of the i/o-interface (–> screen size) is one reason, not to present all available functional interaction point (FIPs) with a specific representation on the screen (RF). So, the user has to navigate through menu structures (= activating DFIPs) to come down to a DC with the desired AFIP. The average length of all possible sequences of dialog operations (PATH) from the top level dialog context down to DCs with the desired AFIP is the basic idea for a good quantitative measure of "interactive directness" (ID): the reciprocal value of the ave-

rage path length (= number of dialog steps). An interface with the maximum ID of 100% has only one DC with path lengths of 1 dialog step. [P = number of all different dialog PATHs]:

$$ID = \left\{ 1/P \sum_{p=1}^{P} lng(PATHp) \right\}^{-1} * 100\% \qquad (2)$$

### 2.3. A quantitative measure of "flexibility" of the dialog interface (DFD)

The number of ways to leave a dialog context (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]:

$$DFD = 1/D \sum_{d=1}^{D} (\#DFIP_d) \qquad (3)$$

### 2.4. A quantitative measure of "flexibility" of the application interface (DFA)

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 (eg. "command" interfaces). [D = number of all different dialog contexts]:

$$DFA = 1/D \sum_{d=1}^{D} (\#AFIP_d) \qquad (4)$$

## 3. VALIDATION OF THE QUANTITATIVE MEASURES

First, a fictive example: we have an application manager with 625 AFIPs. Three different interface types are implemented: (1) a command interface with 1 RF (the "command prompt") and 1 DFIP (e.g. a transparency operator like "ls" of UNIX); (2) a menu interface with a strict hierarchical menu tree (3 levels: 1 DC on level-1 with 5 DFIPs [down], 5 DCs on level-2 with 6 DFIPs [5 down, 1 up] per DC, and 25 DCs on level-3 with 1 DFIP [up] and 5 AFIPs per DC); (3) a desktop interface with a complete net structure (5 DCs with 125 AFIPs and 4 DFIPs per DC). FFB of the menu and the desktop interface is 100%; each f ∈ HF has a r ∈ RF.

Table 1
Comparison of command, menu and desktop interfaces of the fictive example relating to the quantitative measures FFB, ID, DFA, and DFD.

| | #DC | #AFIP | #RAFIP | #DFIP | #RDFIP | FFB% | ID% | DFA | DFD |
|---|---|---|---|---|---|---|---|---|---|
| command | 1 | 625 | 1 | 1 | 0 | 0.2 | 100.0 | 625.0 | 1.0 |
| menu | 31 | 625 | 625 | 60 | 60 | 100.0 | 33.3 | 20.2 | 1.9 |
| desktop | 5 | 625 | 625 | 20 | 20 | 100.0 | 55.6 | 125.0 | 4.0 |

The command interface is the most direct interface (ID=100%) with the highest amount of flexibility (DFA=625), but with more or less no "feedback" (FFB=0.2%). The ID of the desktop interface with the net dialog structure is higher than the ID of menu interface with the hierarchical menu tree.

Second, to validate the quantitative measures (FFB, ID, DFA, DFD) in a more general sense, we need a real product with different interfaces and an empirical investigation. We compared a menu interface (CUI) with a desktop interface (GUI) of the same application manager: a relational data base management system [9]. The results of the product oriented description of

these 2 interfaces with the introduced terms are shown in Table 2. The main result of this empirical investigation is, that the mean task solving time with the GUI is significantly shorter than with the CUI interface (see for more details [9]). One reason for this empirical outcome is the high "interactive directness" (ID=50.5%) and the great "dialog flexibility" (DFD= 21.6) of the GUI. The great influence of the dimension "flexibility" is in accordance with the other results presented in [9]. Interesting is the fact, that the GUI supports the user with less "visual feedback" (FFB= 67%) on average than the CUI (FFB=73%). This high amount of FFB of the CUI is caused by 22 small DCs with FFB= 100%; the GUI has only 14 DCs with FFB=100%.

Table 2
Comparison of a menu and a desktop interface of a relational data base management system relating to the quantitative measures FFB, ID, DFA, and DFD.

| | #DC | #AFIP | #RAFIP | #DFIP | #RDFIP | FFB% | ID% | DFA | DFD |
|---|---|---|---|---|---|---|---|---|---|
| menu | 36 | 434 | 36 | 362 | 161 | 73.0 | 27.7 | 12.1 | 10.1 |
| desktop | 27 | 438 | 38 | 583 | 396 | 67.0 | 50.5 | 16.2 | 21.6 |

## 5. CONCLUSION

Using the 4 quantitative measures for "feedback", "interactive directness" and "flexibility" to measure the interactive quality of user interfaces we are able to classify the most common types: batch, command, menu, desktop. The command 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 interactive directness and high flexibility. The presented approach to quantify usability attributes and the interactive quality of user interfaces 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. In the context of standardization we need criteria to test user interfaces for conformity with standards.

## 6. REFERENCES

[1] 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.

[2] 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.

[3] IFIP (1981): Report of the 1st Meeting of the European User Environment Subgroup of IFIP WF 6.5. German National Center for Computer Science (GMD), P.O. 1316, D-5202 Sankt Augustin (Germany).

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

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

[6] 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.

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

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

[9] 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.

[10] 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.