

HOW TO MEASURE AND TO QUANTIFY USABILITY OF USER INTERFACES

Matthias RAUTERBERG

Work and Organizational Psychology Unit (IfAP), Swiss Federal Institute of Technology (ETH)
Nelkenstrasse 11, CH-8092 Zürich, S WITZERLAND
Tel.: +41-1-6327082, Fax: +41-1-6321186, E-mail: rauterberg@ifap.bepi.ethz.ch

Keywords:

User-interface; analytical method; quantification; metrics

Abstract:

One of the main problems of standards in the context of usability of software quality is, that they can not be measured in product features. We present a new approach to measure user-interface quality in a quantitative way. First, we developed a concept to describe user-interfaces on a granularity level, that is detailed enough to preserve important interface characteristics, and is general enough to cover most of known interface types. We distinguish between different types of 'interaction-points'. With these kinds of interaction-points we can describe several types of interfaces (CUI: command, menu, form-fill-in; GUI: desktop, direct manipulation, multimedia, etc.). We carried out two different comparative usability studies to validate our quantitative measures. The results of one other published comparative usability study can be predicted. Results of six different interfaces are presented and discussed.

INTRODUCTION

We present a new approach to measure user-interface quality in a quantitative way. First, we developed a concept to describe user-interfaces on a granularity level, that is detailed enough to preserve important interface characteristics, and is general enough to cover most of known interface types (command language, CUI, GUI, multimedia, etc.). Different types of user-interfaces can be quantified and distinguished by the general concept of "interaction-points" (IPs). Regarding to the interactive semantic of IPs, different types of IPs must be discriminated.

An interactive system can be distinguished in a dialog and an application manager. So, we distinguish between dialog objects (DO, e.g. "window") and application objects (AO, e.g. "text document"), and dialog functions (DF, e.g. "open window") and application functions (AFIP, e.g. "insert section mark"). Each function $f \in FS$, that changes the state of the content of an application object, is an application function. All other functions are dialog functions (e.g., window operations like move, resize, close). The complete set of all description terms is shown in defined in Table 1.

A dialog context (DC) is defined by all available objects and functions in the actual system state. If the set of available functions changes in the actual DC, then the system changes from one DC to another. In the actual DC all dialog objects (functions, resp.) are perceptible (PO, PF) or hidden (HO, HF). Four different mapping functions relate perceptible structures to hidden objects or functions (see Table 1).

Each interaction-point (IP) is related to at least one interactive function. If both mapping function's δ and α are of the type 1:m(any), then the user-interface is a command interface. 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 (see Figure 1). The perceptual structure (visible, audible, or tactile) of a function (PF) can be, e.g., an icon, earcon, menu option, command prompt, or other mouse sensitive areas. The intersection of PF and PO is sometimes not empty: $PF \cap PO \neq \emptyset$. Icons of graphical interfaces are elements of this intersection, e.g., PDFIP "copy" \equiv PDO "clipboard", PAFIP "delete" \equiv PAO "trash" (see Figure 1). Each interaction-point (IP) is related to at least one interactive function.

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]
$IP := DFIP \cup AFIP$	[interaction-points]
$DFIP := PDFIP \cup HDFIP$	[IPs of dialog functions]
$AFIP := PAFIP \cup HAFIP$	[IPs of application functions]
$\delta :=$ mapping function of a $df \in HDFIP$ to an appropriate $pf \in PF$.	
$\alpha :=$ 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]
$\mu :=$ mapping function of a dialog object $do \in DO$ to an appropriate $po \in PO$.	
$\nu :=$ mapping function of an application object $ao \in AO$ to an appropriate $po \in PO$.	

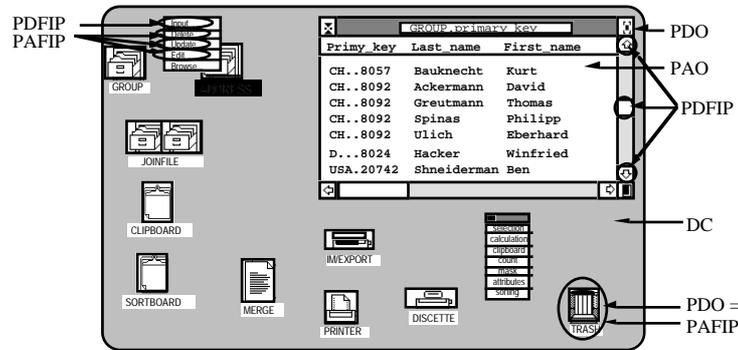


Figure 1. An actual dialog context (DC) of a direct manipulative interface with the representation space of the interactive object (PAO: e.g., data window; PDO: e.g., trash), and the representation space (PF: marked by circles) of the interactive functions (PAFIP: e.g., pop-up menu, trash; PDFIP: e.g., window scrolling).

FOUR QUANTITATIVE MEASURES OF INTERFACE ATTRIBUTES

One important difference between user-interfaces can be the "interactive directness". A user-interface is 100% *interactively direct*, if the user has fully access in the actual dialog context to all $f \in FS$ (see Ulich et al, 1991). This is the case for all command language interfaces. Another important interface attribute is the amount of "feedback". Good interface design is characterised by optimising the multitude of DFIPs (e.g. "flatten" the menu tree, and by allocating an appropriate PDFIP to the remaining HDFIPs. One disadvantage of snapshots (cf. Figure 1) is that all hidden structures could not be referenced. To describe the hidden functionality a schematic view is needed (cf. Rauterberg, 1995).

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 functional feedback" of the function space (FB; see Formula 1). We abbreviate the number of all different dialog contexts with D . A GUI has often a very large number of DCs. To handle this problem we take only all task related DCs into account. Doing this, our measures will give us only a lower estimation for GUIs.

The average length of all possible sequences of interactive operations (PATH) from the top level dialog context (DC, e.g., 'start context') down to DCs with the desired HAFIP or HDFIP can be used as a possible quantitative measure of "interactive directness" (ID, see Formula 2). The measure ID delivers two indices: one for HAFIPs and one for HDFIPs. A PATH has no cycles and

has not more than two additional dialog operations compared with the shortest sequence. An interface with the maximum ID of 100% has only one DC with path lengths of one dialog step. We abbreviate the number of all different dialog paths with P.

Functional feedback:

$$FB = \frac{1}{D} \sum_{d=1}^D (\#PF_d / \#HF_d) * 100\% \quad (1)$$

Interactive directness:

$$ID = \left\{ \frac{1}{P} \sum_{p=1}^P \ln(\text{PATH}_p) \right\}^{-1} * 100\% \quad (2)$$

Application flexibility:

$$DFA = \frac{1}{D} \sum_{d=1}^D (\#HAFIP_d) \quad (3)$$

Dialog flexibility:

$$DFD = \frac{1}{D} \sum_{d=1}^D (\#HDFIP_d) \quad (4)$$

To quantify the flexibility of the application manager we calculate the average number of HAFIPs per dialog context (DFA; see Formula 3). To quantify the flexibility of the dialog manager we calculate the average number of HDFIPs per dialog context (DFD; see Formula 4). A modeless dialog state has maximal flexibility (e.g., "command" interfaces). To interpret the results of our measure's appropriately, empirical studies are necessary.

RESULTS OF APPLYING THE MEASURES

We carried out two different comparative usability studies to validate our measures (experiment-I see Brunner, 1993, and experiment-II see Rauterberg, 1992). A third external comparative study (experiment-III; cf. Grützmaier, 1987) was used for a cross-validation (for a more detailed description see Rauterberg, 1995). All three investigated software products have the same application manager, but two different dialog managers each.

Interesting is the fact, that the GUI of experiment-I supports the user with less "functional feedback" (FB = 66%, see Table 2) on average than the CUI (FB = 73%). This amount of FB of the CUI is caused by 22 small DCs with FB = 100%; the GUI has only 14 DCs with FB = 100%. The amount of functional feedback seems not to be related to the advantage of GUIs. There must be another reason.

Table 2. Comparison our three empirical validation studies relating to the quantitative measures ID, FB, DFA, and DFD. P is the number of all different dialog PATHs for an AFIP or a DFIP; D is the number of all different DCs. ["I1 » I2": interface-1 is better than interface-2]

Experiment	Interface type and dialog structure	D	FB %	ID(AFIP) %	ID(DFIP) %	DFA	DFD	empirical result (p-significance)
I	CUI-hierarchical	36	73	24.7	23.2	12.1	10.1	CUI « GUI p ≤ .001
I	GUI-hierarchical	28	66	22.5	25.5	19.5	20.4	
II	Multimedia-hierarchical	68	100	25.1	28.1	3.6	0.5	MMhier = MMnet p ≤ .085
II	Multimedia-net shaped	65	100	40.7	46.3	4.2	1.3	
III	CUI-hierarchical	363	86	20.9	23.9	2.0	1.9	CUIhier = CUInet p ≤ .825
III	CUI-net shaped	389	90	15.8	21.9	1.3	2.7	

The "interactive directness" is not quite different between both interfaces: CUI: ID = 24.7% for AFIPs and 23.2% for DFIPs versus GUI: ID = 22.5% for AFIPs and 25.5% for DFIPs (see Table 2). Only the two measures of "flexibility" show an important difference: CUI: DFA = 12.1 and DFD = 10.1 versus GUI: DFA = 19.5 and DFD = 20.4 (see Table 2).

In the hierarchical dialog structure (MM_{hier}) of the multimedia information system (experiment-II) only one way is given to reach an AFIP. In the net-shaped version (MM_{net}) several ways are

possible to navigate through the dialog structure. But, what is an AFIP in the context of a multimedia system? We define the *application kernel* of a multimedia system as the set of all masks with a relevant information in the sense of the main purpose of the information system (e.g., concrete information's about bank services in the context of a bank information system); all other masks are part of the dialog manager. A PAFIP is therefore each mouse sensitive area that changes the system to a mask of the application kernel; all mouse sensitive areas are DFIP's.

With the support of Grützmaier we were able to analyse all 752 dialog contexts for both interfaces of the simulation tool 'Moro' (cf. experiment-III). For the hierarchical CUI_{hier} we got the following results: DFA = 2.0 and DFD = 1.9; for the net-shaped CUI_{net}: DFA = 1.3 and DFD = 2.7 (see Table 2). These results for DFA and DFD of both CUI interfaces give us a strong empirical evidence that the following assumptions are correct: (1) The dialog flexibility can be quantitatively measured in a *task independent way*, and (2) the values of DFA and DFD must exceed the *threshold of 15*.

DISCUSSION AND CONCLUSION

If our interpretation of the outcome of experiment-I is correct then we can not find a significant performance difference for dialog structures that remain under the assumed threshold of 15. To control the factor of feedback we carried out the second experiment with a multimedia information system that has 100% functional feedback for both interfaces (Brunner 1993). We picked out a multimedia information system with a hierarchical dialog structure where DFA and DFD are clearly under 15. We implemented a comparable system with a net-shaped dialog structure where DFA and DFD have nearly the same ratio of flexibility as in experiment-I: $DFA_{GUI} / DFA_{CUI} = 1.6$ and $DFA_{MMnet} / DFA_{MMhier} = 1.2$; $DFD_{GUI} / DFD_{CUI} = 2.0$ and $DFD_{MMnet} / DFD_{MMhier} = 2.6$. As we predicted, we could not find a significant performance difference between both types of dialog structures.

To make sure that our results are not biased by our own expectations, we carried out a cross validation study. To do this, (1) we need the outcomes of an external independent comparison study between two different interfaces and (2) the possibility to apply our quantitative measures to all DCs of both interfaces. The empirical investigation of Grützmaier (1987) fulfils both conditions (experiment-III). Given our interpretation of the experiment-I and -II we expected and found a value for DFA and DFD under 15 for experiment-III. We interpret the negative result of experiment-III to the effect that flexibility must exceed our threshold to be effective.

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 standardisation we can use our criteria to test user-interfaces for conformity with standards.

REFERENCES

- BRUNNER, M. AND M. RAUTERBERG: Hierarchische oder netzartige Dialogstruktur bei multimedialen Informationssystemen: eine experimentelle Vergleichsstudie. Technical Report MM-2-93. Institut für Arbeitspsychologie, Zürich: Eidgenössische Technische Hochschule (1993).
- GRÜTZMÄCHER, B.: Datenpräsentation und Lösungsverhalten in einer komplexen, simulierten Problemsituation. Unpublished Master Thesis. (Philosophische Fakultät I, Psychologisches Institut, Abteilung Angewandte Psychologie). Zürich: Universität Zürich (1988).
- RAUTERBERG, M.: 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 (1992).
- RAUTERBERG, M.: 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 (1995).
- ULICH, E., M. RAUTERBERG, T. MOLL, T. GREUTMANN AND O. STROHM: Task orientation and user-oriented dialog design. *International Journal of Human-Computer Interaction* 3(2), 117-144 (1991).