in: Advances in Occupational Ergonomics and Safety I (vol. 1; A. Mital, H. Krueger, S. Kumar, M. Menozzi & J.E. Fernandez, eds.), Cincinnati (Ohio USA): International Society for Occupational Ergonomics and Safety, 1996, pp. 154-157.

How to Measure the Ergonomic Qualitiy of User Interfaces in a Task Independent Way

Matthias RAUTERBERG

Work and Organisation Psychology Unit, Swiss Federal Institute of Technology (ETH)
Nelkenstrasse 11, CH–8092 Zuerich, Email: rauterberg@ifap.bepr.ethz.ch

Abstract: The main problems of standards (e.g., ISO 9241) in the context of usability of software quality are, that they can not measure all relevant product features in a task independent way. 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 (command, menu, form-fill-in, desktop, direct manipulation, multimedia etc.). We analysed the outcomes of three different comparative usability studies to validate our quantitative measures. The results of a published comparative usability study by someone else can be predicted. Results of six different interfaces are presented and discussed. One of the most important result is that the dialog flexibility must exceed a threshold of 15--measured with two of our metrics--to increase significantly the user's performance.

1. Introduction

One of the main problems of standards (e.g., ISO 9241) to quantify software quality of usability are, that they can not measure all relevant product features in a task independent way. To measure interactive qualities four different views on human computer interaction currently exists (see also [1], p. 651; [12]). (1) The *interaction-oriented view:* Usability quality 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. (2) The *user-oriented view:* Usability quality is measured in terms of the mental effort and attitude of the user ("questionnaires" and "interviews"). (3) The *product-oriented view:* Usability quality is measured in terms of the ergonomic attributes of the product itself (quantitative measures). (4) The *formal view:* Usability is formalised and simulated in terms of mental models (formal concepts). Karat [6] describes formal methods in the context of "theory-based" evaluation. The interactive qualities of user interfaces currently are quantified in the context of *interaction-oriented view* and *user-oriented view*, but these both approaches are time consuming and more or less expensive.

2. A descriptive concept of interaction points

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 (*character-oriented user interfaces* CUI: command language, menu, form fill-in; *graphic-oriented user interfaces* GUI: desktop, direct manipulation, multimedia, etc.). Different types of user interfaces can be quantified and distinguished by the general concept of "interaction points". Regarding to the interactive semantic of "interaction points" (IPs), different types of IPs must be discriminated (see also [3]).

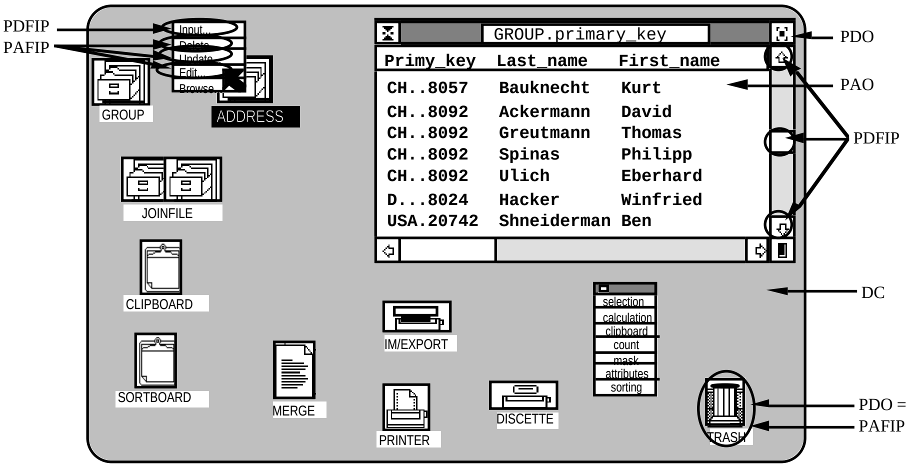
An interactive system can be distinguished in a dialog and an application manager. So, we distinguish between dialog objects (DO, e.g. "window") and application objects (AO, e.g. "text document"), and dialog functions (DF, e.g. "open window") and application functions (AF, e.g. "insert section mark"). Each function $f \in FS$, that changes the state 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 defined as follows: *interaction space* $IS := OS \times FS$; *dialog context* $DC \in IS$; *object space* $OS := PO \cup HO$; *function space* $FS := PF \cup HF$; *(perceptible) representations of objects* $PO := PDO \cup PAO$; *hidden objects* $HO := HDO \cup HAO$; *(perceptible) representations of functions* $PF := PDFIP \cup PAFIP$; *hidden functions* $HF := HDFIP \cup HAFIP$; *(perceptible) represented dialog function points* $PDFIP := \{(df,pf) \in HDFIP \times PF: pf = \delta(df)\}$; *(perceptible) represented application function points* $PAFIP := \{(af,pf) \in HAFIP \times PF: pf = \alpha(af)\}$; *interaction points* $IP := DFIP \cup AFIP$; *interaction points of dialog functions* $DFIP := PDFIP \cup HDFIP$; *interaction points of application functions* $AFIP := PAFIP \cup HAFIP$; 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 $pf \in PF$; *(perceptible) represented dialog objects* $PDO := \{(do,po) \in HDO \times PO: po = \mu(do)\}$; *(perceptible) represented application objects*

PAO := {(ao,po) ∊ HAO x PO: po = ν(ao)}; μ := mapping function of a dialog object do ∊ DO to an appropriate po ∊ PO; ν := mapping function of an application object ao ∊ AO to an appropriate po ∊ PO.

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

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 ∊ FS is related to a perceptible structure PF. 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 ∩ PO ≠ ∅. In the context of graphical interfaces icons are elements of this intersection, e.g., PDFIP "copy" ≡ PDO "clipboard", PAFIP "delete" ≡ PAO "trash" (see Figure 1). Each interaction point (IP) is related to at least one interactive function.



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 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).

One important difference between a menu and a direct manipulative interface 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 ∊ FS (cf. [7]). Good interface design is characterised by optimising the multitude of DFIPs (e.g. "flatten" the menu tree; see [8]) and by allocating an appropriate PDFIP to the remaining HDFIPs.

3. Quantitative measures of user interface characteristics

To estimate the amount of functional feedback of an interface the following 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 a lower estimation for GUIs.

Formula 1: Functional feedback:

$$FB = \frac{1}{D}\sum_{d=1}^{D}\left(\#\,\mathrm{PF}_d/\#\,\mathrm{HF}_d\right)*100\%$$

Formula 2: Interactive directness:

$$ID = \left\{\frac{1}{P}\sum_{p=1}^{P}\mathrm{lng}\left(\mathrm{PATH}_p\right)\right\}^{-1} *100\%$$

Formula 3: Application flexibility:

$$DFA = \frac{1}{D}\sum_{d=1}^{D}\left(\#\,HAFIP_d\right)$$

Formula 4: Dialog flexibility:

$$DFD = \frac{1}{D}\sum_{d=1}^{D}\left(\#\,HDFIP_d\right)$$

The average length (lng) 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.

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 measures appropriately, we need empirical studies.

4. The empirical validation

We carried out two different comparative usability studies to validate our measures [9] [2]. A third external comparative study [5] was used for a cross validation. All three investigated software products have the same application manager, but two different dialog managers each.

4.1. Results and discussion of experiment-I

We [9] compared an old, ascii-based CUI-interface (menu) of a relational database management system with a new GUI-interface (desktop). The main result of this empirical investigation was, that the mean task solving time with the GUI is significantly shorter than with the CUI interface (see last column in Table 1). How can we explain this difference? Our first interpretation of this outcome was the supposed different amount of 'transparency' [13], because one aspect of 'transparency' is 'feedback' (see [4], pp. 318-321).

Interesting is the fact, that the GUI supports the user with less "functional feedback" (FB = 66%, see fourth column in Table 1) 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 to explain the performance difference between CUI and GUI.

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 1). 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 1). We interpret this result to the effect that flexibility must exceed a threshold to be effective (DFD, DFA > 15).

Table 1. 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-2 is better than interface-1]

Experi ment	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 $p \leq .001$
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	$MM_{hier} = MM_{net}$ $p \leq .085$
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	$CUI_{hier} = CUI_{net}$ $p \leq .825$
III	CUI-net shaped	389	90	15.8	21.9	1.3	2.7	

4.2. Results and discussion of experiment-II

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 a second experiment with a multimedia information system that has 100% functional feedback for both interfaces [2]. 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 had 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 can not find a significant performance difference between both types of dialog structures (see Table 1). 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 Grützmacher [5] fulfiled both conditions.

4.3. Results and discussion of experiment-III

The study of Grützmacher [5] was carried out to investigate research questions in the context of how to control a complex domain with a simulation tool. One independent factor was varied: the dialog structure (hierarchical versus net-shaped). This simulation tool was implemented on a mainframe computer system with character-oriented terminals (IBM 3270). The dependent variable was not 'task solving time' but 'target discrepancy' as a performance measure. The sample consists of 20 users with the hierarchical dialog structure and 15 users with the net-shaped structure. The main result was that the factor 'dialog structure' did not show a significant difference. Given our interpretation of the last two experiments we expected a value for DFA and DFD under 15. With the generous support of Grützmacher we were able to analyse all 752 dialog contexts for both interfaces. For the hierarchical CUI we got the following results (see last two rows in Table 1): DFA = 2.0 and DFD = 1.9 and for the net-shaped CUI: DFA = 1.3 and DFD = 2.7. These results for DFA and DFD of both CUI interfaces give us a sufficient evidence that the following assumptions seem to be correct: (1) We can measure the dialog flexibility in a task independent and quantitative way, and (2) the values of DFA and DFD must exceed the threshold of 15.

5. Conclusion

Using the four 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: command, menu, desktop (see [10]). The command interface is characterised by high interactive directness, but this interface type has a very low amount of visual feedback. Especially graphical interfaces (e.g., multimedia) can support users with sufficient interactive directness. GUIs are characterised by high dialog 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 standardisation we can use our criteria to test user interfaces for conformity with standards. A more detailed description of all introduced terms and further applications are given in Rauterberg [11].

References

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

[2] Brunner, M. and M. Rauterberg (1993). *Hierarchische oder netzartige Dialogstruktur bei multimedialen Informationsystemen: eine experimentelle Vergleichsstudie.* Technical Report MM-2-93. Institut für Arbeitspsychologie, Eidgenössische Technische Hochschule, Zürich.

[3] Denert, E. (1977). Specification and design of dialogue systems with state diagrams. In: *International Computing Symposium 1977* (E. Morlet and D. Ribbens, eds.), 417-424. North-Holland.

[4] Dix, A., J. Finlay, G. Abowd and R. Beale (1993). *Human-Computer Interaction*. Prentice Hall.

[5] Grützmacher, B. (1988). *Datenpräsentation und Lösungsverhalten in einer komplexen, simulierten Problemsituation.* Unpublished Master Thesis. (Philosophische Fakultät I, Psychologisches Institut, Abteilung Angewandte Psychologie). Universität Zürich, Zürich.

[6] Karat, J. (1988). Software Evaluation Methodologies. In: *Handbook of Human-Computer Interaction* (M. Helander, ed.), 891-903. Elsevier.

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

[8] Paap, K. and R. Roske-Hofstrand (1988). Design of menus. In: *Handbook of Human-Computer Interaction* (M. Helander, ed.), 205-235. Elsevier.

[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] Rauterberg, M. (1993). Quantitative Measures to Evaluate Human-Computer Interfaces. In: *Human-Computer Interaction: Applications and Case Studies* (M. Smith and G. Salvendy, eds.), Advances in Human Factors/ Ergonomics Vol. 19A), 612-617. Elsevier.

[11] Rauterberg, M. (1995) *Ein Konzept zur Quantifizierung software-ergonomischer Richtlinien*. (Doctoral Dissertation), Institut für Arbeitspsychologie, Eidgenössische Technische Hochschule, Zürich.

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

[13] Ulich, E., M. Rauterberg, T. Moll, T. Greutmann and O. Strohm (1991). Task orientation and user-oriented dialog design. *International Journal of Human-Computer Interaction* 3(2), 117-144.