How to Measure the Ergonomic Quality of User Interfaces in a Task Independent Way
Rauterberg, G.W.M.

Published in:
Advances in occupational ergonomics and safety I

Published: 01/01/1996

Document Version
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the author's version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

Citation for published version (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal?

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.
How to Measure the Ergonomic Quality 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 ∈ 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 x FS; dialog context DC ∈ IS; object space OS := PO ∪ HO; function space FS := PF ∪ HF; (perceptible) representations of objects PO := PDO ∪ PAO; hidden objects HO := HDO ∪ HAO; (perceptible) representations of functions PF := PDFIP ∪ PAFIP; hidden functions HF := HDFIP ∪ HAFIP; (perceptible) represented dialog function points PDFIP := {(df,pf) ∈ HDFIP x PF; pf = δ(df)}; (perceptible) represented application function points PAFIP := {(af,pf) ∈ HAFIP x PF; pf = α(af)}; interaction points IP := DFIP ∪ AFIP; interaction points of dialog functions DFIP := PDFIP ∪ HDFIP; interaction points of application functions AFIP := PAFIP ∪ HAFIP; δ := mapping function of a df ∈ HDFIP to an appropriate pf ∈ PF; α := mapping function of an af ∈ HAFIP to an appropriate pf ∈ PF; (perceptible) represented dialog objects PDO := {(do,po) ∈ HDO x PO; po = µ(do)}; (perceptible) represented application objects PADO := {(ao,po) ∈ HAO x PO; po = μ(ao)}; (perceptible) represented hidden objects HDO := {ho ∈ HDO; ho = μ(ho)}; (perceptible) represented hidden functions HDFIP := {(df,df) ∈ HDFIP x HDFIP; δ(df) = df}.
PAO := \{(ao, po) \in HAO \times PO: po = \nu(ao)\}; \mu := \text{mapping function of an application object } ao \in AO \text{ to an appropriate po} \in 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 \(d\) and \(a\) are of the type 1:m(any), then the user interface is a command interface. If both mapping function’s \(d\) and \(a\) 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. 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 \(\neq\) \(\varnothing\). In the context of graphical interfaces icons 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.

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 \(f\) \in 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:**

\[
FB = \frac{1}{D} \sum_{d=1}^{D} \left(\frac{\#PF_d}{\#HF_d}\right) \times 100\%
\]

**Formula 2: Interactive directness:**

\[
ID = \left\{ \frac{1}{P} \sum_{p=1}^{P} \ln\left(\frac{\text{PATH}_p}{\text{PATH}_p}\right) \right\}^{-1} \times 100\%
\]

**Formula 3: Application flexibility:**

\[
DFA = \frac{1}{D} \sum_{d=1}^{D} (\#HAFIP_d)
\]

**Formula 4: Dialog flexibility:**

\[
DFD = \frac{1}{D} \sum_{d=1}^{D} (\#HDFIP_d)
\]
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]

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

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: DFDGUIMM / DFDGUICUI = 1.6 and DFAMMhier / DFAMMhier = 1.2; DFDGUIMM / DFDGUICUI = 2.0 and DFAMMnet / DFAMhier = 2.6.
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 interface (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