How to measure and to quantify usability attributes of man-machine interfaces
Rauterberg, G.W.M.

Published in:
RO-MAN: robot and human communication: IEEE international workshop, 5th, November 11-14, 1996, Tsukuba

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 and to Quantify Usability Attributes of Man-Machine Interfaces

M. Rauterberg

Work and Organisation Psychology Unit (IfAP)
Swiss Federal Institute of Technology (ETH)
Nelkenstrasse 11, CH-8092 ZURICH, Switzerland
URL: http://www.ifap.bepr.ethz.ch/~rauterberg

Abstract

Several views on man-machine interaction are possible to measure 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: perceptible versus hidden, and dialog versus application oriented. 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.

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 [22]; [3], p. 651).

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 [11].

The user-oriented view: usability is measured in terms of the mental effort and attitude of the user (“questionnaires” and “interviews”).

The formal view: usability is formalised and simulated in terms of mental models (formal concepts). Karat [10] describes formal methods in the context of “theory-based” evaluation.

The product-oriented view: usability is measured in terms of the ergonomic attributes of the product (quantitative measures). All heuristic evaluations carried out by ergonomic experts investigating a concrete product fall in this category, too [8].

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. Usability testing is constrained to the investigated task solving processes and the selected users, too. It would be 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.

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 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 functions (PF) and hidden functions (HF). A concrete dialog context (DC) contains a subset of {OS ∪ FS}.

An interactive system can be distinguished in a dialog and an application manager [6]. 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 [5] 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 (Figure 1).
A perceptible AFIP is called a PAFIP and a perceptible DFIP is called a PDFIP (see Figure 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 as follows (for a more detailed version see [21]):

$$\delta = \frac{\#PF}{\#HF} \times 100\%$$

$$\alpha = \frac{\#PF}{\#HF} \times 100\%$$

If both mapping function’s $\delta$ and $\alpha$ are of the type 1 to m(any), then the user interface is a command interface (see Figure 2) where the command interface has only one p=1 PF, the "command prompt" (e.g. the PF in Figure 1). If both mapping function’s $\delta$ and $\alpha$ are of the type 1:1, then the user interface is a menu or direct manipulative interface where each f=1 FS is related to a perceptible structure PF 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% interactivly direct, if the user has fully access in the actual dialog context to all AFIPs [12]. Good interface design is characterized by optimising the multitude of DFIPs (e.g. "flatten" the menu tree [17]) and by allocating an appropriate PDFIP to the remaining HDFIPs.

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 [21]). If each functional interaction point (FIP) has its own representational interaction point (PF), then the user has 100% feedback (FB) 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 HF" (#HF = #HDFIP + #HAFIP) per dialog context. This ratio quantifies the average "amount of feedback" of the function space (FB). (D is the number of all different dialog contexts.)

**Functional feedback:**

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

**Interactive Directness:**

$$ID = \left\{ \frac{1}{P} \sum_{p=1}^{P} \frac{\text{lng} (PATH_p)}{\text{lng} (PATH_p)} \right\} - 1 \times 100\%$$

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. [21]). 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.)

3 Description of Interfaces

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 [27]. These techniques can be summarised into three different interaction styles:

**Command [language] interfaces (CI):**

This interaction style by typing in words from a set of legal commands is one of the oldest ways 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 $\text{ID} = 1$ (for examples see [21]).

**Cons:** The user has no permanent feedback of all actual available function points. This aspect can be measured with the metric $\text{FB} < 1$ (see Table 1).

**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 $\text{FB} = 100\%$ (see Table 1).

**Cons:** Finding a function point in deeper menu hierarchies is cumbersome; this can be measured with the metric $\text{ID} = 1$ (for example see Figure 2).

**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 [21]).

**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.

### 4 Classification of 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 1). 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. [20]).

To make this classification 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.

**Table 1.** A classification schema of most common user interfaces.

<table>
<thead>
<tr>
<th>[visual] feedback (FB)</th>
<th>low</th>
<th>high</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>interactive directness (ID)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>low</td>
<td>batch</td>
<td>menu interface MI</td>
</tr>
<tr>
<td>high</td>
<td>command language CI</td>
<td>direct manipulation DI</td>
</tr>
</tbody>
</table>

### 5 Empirical validation of the classification

A major task in our area of HCI is the development of a theoretical explanation of the outcomes presented in Figure 1 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 (see Figure 1). 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 2). 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).

The general result of this first overview (Table 2) 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 3). 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 3) 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 2. 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)

<table>
<thead>
<tr>
<th>Reference</th>
<th>Interface</th>
<th>Skill Level</th>
<th>Usability Metric</th>
<th>Outcome</th>
<th>Test Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Streitz et al. (1987)</td>
<td>CI, MI</td>
<td>beginner</td>
<td>task solving time</td>
<td>CI &lt; MI</td>
<td>sig.</td>
</tr>
<tr>
<td>Chin et al. (1988)</td>
<td>CI, MI</td>
<td>beginner</td>
<td>subjective rating</td>
<td>CI &lt; MI</td>
<td>sig.</td>
</tr>
<tr>
<td>Ogden &amp; Boyle (1982)</td>
<td>CI, MI, HY</td>
<td>beginner</td>
<td>preferences</td>
<td>CI &lt; MI</td>
<td>sig.</td>
</tr>
<tr>
<td>Roy (1992)</td>
<td>CI, MI</td>
<td>advanced</td>
<td>error rate</td>
<td>CI &lt; MI</td>
<td>sig.</td>
</tr>
<tr>
<td>Roberts &amp; Moran (1983)</td>
<td>CI, MI, DI</td>
<td>experts</td>
<td>task solving time</td>
<td>CI &lt; MI</td>
<td>sig.</td>
</tr>
<tr>
<td>Chin et al. (1988)</td>
<td>CI, MI</td>
<td>experts</td>
<td>subjective rating</td>
<td>CI &lt; MI</td>
<td>sig.</td>
</tr>
<tr>
<td>Peters et al. (1990)</td>
<td>CI, MI, DI</td>
<td>experts</td>
<td>slips</td>
<td>CI &lt; MI</td>
<td>sig.</td>
</tr>
<tr>
<td>Peters et al. (1990)</td>
<td>CI, MI, DI</td>
<td>experts</td>
<td>recognition errors</td>
<td>CI &lt; MI</td>
<td>sig.</td>
</tr>
<tr>
<td>Peters et al. (1990)</td>
<td>CI, MI, DI</td>
<td>experts</td>
<td>efficiency</td>
<td>CI &lt; MI</td>
<td>sig.</td>
</tr>
<tr>
<td>Ogden &amp; Boyle (1982)</td>
<td>CI, MI, HY</td>
<td>beginner</td>
<td>task time</td>
<td>CI &lt; MI</td>
<td>not sig.</td>
</tr>
<tr>
<td>Roy (1992)</td>
<td>CI, MI</td>
<td>advanced</td>
<td>task solving time</td>
<td>CI &lt; MI</td>
<td>not sig.</td>
</tr>
<tr>
<td>Antin (1988)</td>
<td>CI, MI, KMI</td>
<td>advanced</td>
<td>subjective rating</td>
<td>CI &lt; MI</td>
<td>not sig.</td>
</tr>
<tr>
<td>Hauptmann &amp; Green (1983)</td>
<td>CI, MI, NO</td>
<td>beginner</td>
<td>task solving time</td>
<td>CI = MI</td>
<td>not sig.</td>
</tr>
<tr>
<td>Hauptmann &amp; Green (1983)</td>
<td>CI, MI, NO</td>
<td>beginner</td>
<td>number of errors</td>
<td>CI = MI</td>
<td>not sig.</td>
</tr>
<tr>
<td>Hauptmann &amp; Green (1983)</td>
<td>CI, MI, NO</td>
<td>beginner</td>
<td>subjective rating</td>
<td>CI = MI</td>
<td>not sig.</td>
</tr>
<tr>
<td>Whiteside et al. (1985)</td>
<td>CI, MI, IO</td>
<td>beginner</td>
<td>task completion rate</td>
<td>CI &gt; MI</td>
<td>not sig.</td>
</tr>
<tr>
<td>Antin (1988)</td>
<td>CI, MI, KMI</td>
<td>advanced</td>
<td>preferences</td>
<td>CI &gt; MI</td>
<td>not sig.</td>
</tr>
<tr>
<td>Roberts &amp; Moran (1983)</td>
<td>CI, MI, DI</td>
<td>experts</td>
<td>error-free task time</td>
<td>CI &gt; MI</td>
<td>not sig.</td>
</tr>
<tr>
<td>Whiteside et al. (1985)</td>
<td>CI, MI, IO</td>
<td>advanced</td>
<td>task completion rate</td>
<td>CI &gt; MI</td>
<td>sig.</td>
</tr>
<tr>
<td>Streitz et al. (1987)</td>
<td>CI, MI</td>
<td>advanced</td>
<td>task solving time</td>
<td>CI &gt; MI</td>
<td>sig.</td>
</tr>
<tr>
<td>Antin (1988)</td>
<td>CI, MI, KMI</td>
<td>advanced</td>
<td>task completion rate</td>
<td>CI &gt; MI</td>
<td>sig.</td>
</tr>
<tr>
<td>Whiteside et al. (1985)</td>
<td>CI, MI, IO</td>
<td>experts</td>
<td>task completion rate</td>
<td>CI &gt; MI</td>
<td>sig.</td>
</tr>
</tbody>
</table>

Table 3. 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)

<table>
<thead>
<tr>
<th>Reference</th>
<th>Interface</th>
<th>Skill Level</th>
<th>Usability Metric</th>
<th>Outcome</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Altmann (1987)</td>
<td>CI, DI</td>
<td>beginner</td>
<td>task solving time</td>
<td>CI &lt; DI</td>
<td>sig.</td>
</tr>
<tr>
<td>Karat et al. (1987)</td>
<td>CI, DI</td>
<td>beginner</td>
<td>subjective rating</td>
<td>CI &lt; DI</td>
<td>sig.</td>
</tr>
<tr>
<td>Streitz et al. (1989)</td>
<td>CI, DI</td>
<td>beginner</td>
<td>task solving time</td>
<td>CI &lt; DI</td>
<td>sig.</td>
</tr>
<tr>
<td>Sengupta &amp; Te’eni (1991)</td>
<td>CI, DI</td>
<td>beginner</td>
<td>task solving time</td>
<td>CI &lt; DI</td>
<td>sig.</td>
</tr>
<tr>
<td>Margono et al. (1987)</td>
<td>CI, DI</td>
<td>beginner</td>
<td>number of errors</td>
<td>CI &lt; DI</td>
<td>sig.</td>
</tr>
<tr>
<td>Morgan et al. (1991)</td>
<td>CI, DI</td>
<td>beginner</td>
<td>time between errors</td>
<td>CI &lt; DI</td>
<td>sig.</td>
</tr>
<tr>
<td>Morgan et al. (1991)</td>
<td>CI, DI</td>
<td>beginner</td>
<td>error correction time</td>
<td>CI &lt; DI</td>
<td>sig.</td>
</tr>
<tr>
<td>Morgan et al. (1991)</td>
<td>CI, DI</td>
<td>beginner</td>
<td>error-free time</td>
<td>CI &lt; DI</td>
<td>sig.</td>
</tr>
<tr>
<td>Morgan et al. (1987)</td>
<td>CI, DI</td>
<td>beginner</td>
<td>subjective rating</td>
<td>CI &lt; DI</td>
<td>sig.</td>
</tr>
<tr>
<td>Morgan et al. (1991)</td>
<td>CI, DI</td>
<td>beginner</td>
<td>subjective rating</td>
<td>CI &gt; DI</td>
<td>sig.</td>
</tr>
<tr>
<td>Torres-Chazaro et al.(1992)</td>
<td>CI, DI</td>
<td>beginner</td>
<td>subjective rating</td>
<td>CI &lt; DI</td>
<td>sig.</td>
</tr>
<tr>
<td>Sengupta &amp; Te’eni (1991)</td>
<td>CI, DI</td>
<td>beginner</td>
<td>efficient usage</td>
<td>CI &lt; DI</td>
<td>sig.</td>
</tr>
<tr>
<td>Tombaugh et al. (1989)</td>
<td>CI, DI</td>
<td>advanced</td>
<td>subjective rating</td>
<td>CI &lt; DI</td>
<td>sig.</td>
</tr>
<tr>
<td>Torres-Chazaro et al.(1992)</td>
<td>CI, DI</td>
<td>advanced</td>
<td>subjective rating</td>
<td>CI &lt; DI</td>
<td>sig.</td>
</tr>
<tr>
<td>Roberts &amp; Moran (1983)</td>
<td>CI, MI, DI</td>
<td>experts</td>
<td>task solving time</td>
<td>CI &lt; DI</td>
<td>sig.</td>
</tr>
<tr>
<td>Peters et al. (1990)</td>
<td>CI, MI, DI</td>
<td>experts</td>
<td>oblivion’s errors</td>
<td>CI &lt; DI</td>
<td>sig.</td>
</tr>
<tr>
<td>Peters et al. (1990)</td>
<td>CI, MI, DI</td>
<td>experts</td>
<td>recognition error</td>
<td>CI &lt; DI</td>
<td>sig.</td>
</tr>
<tr>
<td>Peters et al. (1990)</td>
<td>CI, MI, DI</td>
<td>experts</td>
<td>efficiency</td>
<td>CI &lt; DI</td>
<td>sig.</td>
</tr>
<tr>
<td>Margono et al. (1987)</td>
<td>CI, DI</td>
<td>beginner</td>
<td>task solving time</td>
<td>CI &lt; DI</td>
<td>not sig.</td>
</tr>
<tr>
<td>Morgan et al. (1991)</td>
<td>CI, DI</td>
<td>beginner</td>
<td>task solving time</td>
<td>CI &lt; DI</td>
<td>not sig.</td>
</tr>
<tr>
<td>Tombaugh et al. (1989)</td>
<td>CI, DI</td>
<td>advanced</td>
<td>task solving time</td>
<td>CI &lt; DI</td>
<td>not sig.</td>
</tr>
<tr>
<td>Roberts &amp; Moran (1983)</td>
<td>CI, MI, DI</td>
<td>experts</td>
<td>error correction time</td>
<td>CI &lt; DI</td>
<td>not sig.</td>
</tr>
<tr>
<td>Altmann (1987)</td>
<td>CI, DI</td>
<td>beginner</td>
<td>subjective rating</td>
<td>CI &gt; DI</td>
<td>not sig.</td>
</tr>
<tr>
<td>Masson et al. (1988)</td>
<td>CI, DI</td>
<td>advanced</td>
<td>task solving time</td>
<td>CI &gt; DI</td>
<td>sig.</td>
</tr>
</tbody>
</table>

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 ([17], p.207): Two or more commercially available systems are compared, which have different application managers (e.g.: [32] [1]), 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., [28]). Another problem seems to be the selection of real expert users. So
normally empirical investigations are done with beginners only (e.g.: [13] [29]), 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--, this aspect should be considered, too.

If the classification of the three most common interfaces in chapter 2 is valid, 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 incongruent and inconsistent results in Table 2.

One of the main goals 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 qualitative 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 [24].

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. We can find a significant correlation between both dimensions (p ≤ .044; see Table 4). This correlation means that CI has a higher chance of 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 1) is one possible and plausible interpretation. Therefore, we interpret this result as an empirical validation of our two metrics FB and ID.

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"). We can find a significant correlation between both dimensions (p ≤ .018; see Table 5). This correlation means that the outcome "CI better as MI or DI" can be significantly more often observed with advanced users than with beginners. This result is a first empirical confirmation of the often expressed opinion that CI is especially good for experts.

Table 4. Contingency table of a meta-analysis only for significant differences (result: $\text{Chi}^2 = 4.07$, df = 1, p ≤ .044).

<table>
<thead>
<tr>
<th></th>
<th>MI</th>
<th>DI</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI better as ...</td>
<td>4 (2.0)</td>
<td>1 (3.0)</td>
</tr>
<tr>
<td>CI worse as ...</td>
<td>9 (11.0)</td>
<td>19 (17.0)</td>
</tr>
</tbody>
</table>

Table 5. Contingency table of a meta-analysis only for significant differences (result: $\text{Chi}^2 = 5.55$, df = 1, p ≤ .018).

<table>
<thead>
<tr>
<th></th>
<th>beginner</th>
<th>advanced+</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI better as MI, DI</td>
<td>0 (2.4)</td>
<td>5 (2.6)</td>
</tr>
<tr>
<td>CI worse as MI, DI</td>
<td>16 (13.6)</td>
<td>12 (14.4)</td>
</tr>
</tbody>
</table>

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" (FB) 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 graphic 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" [20]. 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 [21]).

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 empiri
References


