Unifying functional User Interface design principles

Jenny Ruiz^{[ID](#)^a}, E. Serral^{[ID](#)^{b*}} and M. Snoeck^{[ID](#)^b}

^a*Faculty of Informatics and Mathematics, University of Holguin, Holguin, Cuba;*

^b*Faculty of Business and Economics, KU Leuven, Leuven, Belgium*

Warmoesberg 26 - 1000 Brussel, Belgium, +32 16 32 68 79,

estefania.serralasensio@kuleuven.be

Many designers and User Interface (UI) educators discuss principles to be followed when designing the functional aspects of a UI. However, many UI principles have been proposed, scattered in scientific papers and teaching books. Some principles are different, and many are somehow similar or overlapping with others. This makes it very difficult to comprehend where to pay attention to when designing a UI. In this paper, we perform a systematic literature review to first identify the most relevant authors in the domain of functional UI design principles. Focusing on the three most cited works of these authors, we extracted 257 principles. We next analyzed all these principles, unified their variants, and, considering their scientific influence, finally derived a shorter and core selection of 36 principles. This core selection provides educators and UI designers with a clear path to teach, evaluate, learn and improve the UI functional design.

Keywords: interaction design; user interface design; design principles

1. Introduction

Today's Information Technology (IT) industry is characterized by a heterogeneity of platforms, devices, modalities, programming languages, working environments, and users with different characteristics and skills, among others. This heterogeneity represents a huge challenge that forces us to create highly complex software applications with a large variety of features.

Heterogeneity of platforms and user expectations, e.g. in terms of user friendliness, have turned the User Interface (UI) of an application into one of the most important elements to consider when developing applications since it connects the end-

users to the functionality of the software (Akiki et al., 2015; Hentati et al., 2016; Martins & Garcia, 2015; Molina et al., 2012; Myers et al., 2000; Shneiderman, 1997; Yigitbas et al., 2017).

The importance of the UI as part of an application makes it extremely important that software developers know how to design a UI. Nevertheless, developers mostly fall short when it comes to aspects of usability engineering as part of the software engineering processes (Polack-Wahl, 2004). As the success or failure of a software application depends on its usability (Molina et al., 2012; van Schaik & Ling, 2008), poor knowledge on issues of usability leads to little or no appreciation for UI issues and related constraints on the functional core (Adikari et al., 2009).

As explained by (Galitz, 2007), a proper interface design should satisfy the users' needs, capabilities and limitations. Even when different systems have different kinds of users, there are some inherent and similar characteristics in the way the users interact with the applications as humans. The study of this human behavior with computers has led to the generalization of some design principles that help to design usable UIs. These "UI design principles" encompass the best practices in UI design, agreed upon by most experts in the field. These principles present high level concepts and beliefs that allow guiding the software design (Mandel, 1997), while having a positive effect on usability (Folmer & Bosch, 2004). Incorporating UI design principles results in better performance and increased acceptance from users (Estock et al., 2018).

Many authors have defined design principles, e.g., (Mandel, 1997; Nielsen, 1994c; Norman, 1983b; Shneiderman & Plaisant, 2005). How the principles are defined is however different per author, e.g., principles may be defined at different levels of abstraction and may or may not contain guidelines to implement them. Also, some principles are proposed by several authors, while others are proposed by only one.

In the literature, only a few studies can be found that analyze UI design principles. Kimball (2013) presents a study focused on visual design principles like balance, contrast, and alignment. In this study, principles are searched in books and web-sites on design and designers, design educators and design students are surveyed on how the principles are related to one another. Ahmed (2010) presents a comparative study of the design elements and principles proposed by several authors. This author proposes balance, rhythm, proportion and emphasis as the basic design principles, stating all the other principles are only variations of these four design principles. Finally, a conceptual framework for a systematic mapping of layout design principles based on the Yin and Yang theory is proposed in (Park, 2009). These three studies focus on visual design principles and tackle, amongst others, the aesthetic and artistic appeal related to graphic design. The graphic design is however like a skin that covers functionalities. Obviously, the functional aspects under this skin are equally important when it comes to usability.

There are other studies for functional design principles that are very domain or purpose specific and cannot easily be generalized. Examples are the works proposed by (Chorianopoulos, 2008), (for interactive television applications), (Petrovčič et al., 2018), (for elderly users), (Min & Lee, 2019), (for infrequent tasks). However, as far as we know, there are no studies that compile UI design principles from a functional perspective with a wider scope. In the remainder of this paper, we will use the term "functional UI design principles" or "UI design principles" for short.

This lack of clear set of agreed upon functional UI design principles is problematic. As explained above, it is of utmost importance to train software engineers in UI design. Yet, training in design involves learning about principles (Kimball, 2013),

but the wide variety of design principles can make it very difficult for new designer to understand where their focus should be.

While general design principles or heuristics exist for graphic user interfaces, “it remains a challenge to facilitate the implementation of these traditional interface interaction heuristics for design” (Miller et al., 2018). Even though they exist, design principles or heuristics are not yet used enough and are difficult to understand by software developers (Seffah & Metzker, 2004). Providing them with support for learning UI design principles can optimize UI design.

The goal of this paper is to address this gap by presenting a compilation of core UI design principles related to functional aspects. Figure 1 shows the steps of the overall research method for the collection, analysis and selection of UI design principles. For each step, the detailed method is explained in the corresponding section: steps 1 and 2 are described in Section 2, and steps 3 and 4 are described in Sections 3 and 4 respectively.

Figure 1. Method followed for the analysis and selection of UI design principles.

The first step is a Systematic Literature Review (SLR) on UI design principles (see Section 2.1). The texts discovered by the SLR include books, UI design courses, scholarly articles and web sites that focus on UI design, design principles, UI principles and similar subjects. The outcome of this step is a set of most relevant authors. In the

second step, the design principles were compiled: we first compiled the set of most relevant works of the most relevant authors in this domain and the UI design principles proposed in these works (see Sections 2.2 and 2.3). Then, due to the fact that many variants from the same principles exist and that some were duplicated or overlapping each other, a reduction of the principles was made (see Section 3). In the last step, we analyzed the UI design principles that are most referenced in the literature taking into account all the principles' variants and articles where they are published (see Section 4). Section 5 discusses the lessons learned-and the limitations of this research. Finally, Section 6 concludes the paper.

2. Principles proposed in the literature (steps one and two)

The goal of the first step is "to collect all principles". As identifying principles cannot be done without identifying first the works in which they are presented, we collected the principles in two steps: 1) identifying all relevant authors, and 2) identifying all the principles proposed by each author. Each of these steps is explained in detail below, including how difficulties in the identification process were dealt with.

2.1 Identifying relevant authors and their works

Identifying all relevant papers or books that present design principles is complicated by the fact that there is no unique term to coin design principles. As (Scapin & Bastien, 1997) posit, there are many different ways to capture human factor knowledge associated to UI design. This knowledge may not always be called "design principles" but certainly can be identified as good practices. Some examples are "golden rules" (Shneiderman, 1997), sets of guidelines (Smith & Mosier, 1986) and standards (from organizations like ISO).

The use of a search term in a database would therefore always only yield a subset of the relevant works. In such a case, snowballing may be an important step to

complete an initial set of papers. Snowballing can be done by looking at forward and backward citations. However, many papers would cite a paper that refers to UI design principles, but without presenting guidelines itself. Also, each found paper can cite many more papers (and not only one paper) to refer to a principle, its use, or an author. Hence, snowballing through the references would lead to a massive amount of papers to go through, many of which would be irrelevant for our purpose. For that reason, looking up the other works of the same authors would be a better approach. For other problems we faced during the process of identifying the relevant authors and their works the reader is referred to subsection 5.1.

So, to identify the relevant works we first need to identify the relevant authors in this domain. To that end we performed a SLR, following the guidelines proposed by (Kitchenham & Charters, 2007). These guidelines propose to perform the SLR in three phases: planning, conducting, and reporting.

Planning the SLR

To answer the research question "Which authors have proposed UI design principles?", we searched the Scopus database, which is one of the largest databases of peer-reviewed literature. User Interfaces is a term used in software and technologies of any field of knowledge. However, its development is inherent to the fields of computer science, informatics, and engineering. As our goal is to unify UI functional design principles, we decided to focus on these fields and exclude papers in others like mathematics, medicine, business, etc. We therefore used the following search string:

TITLE-ABS-KEY ("user interface" AND ("design principles" OR "golden rules"))
AND (EXCLUDE (SUBJAREA , "MATH") OR EXCLUDE (SUBJAREA ,
"SOCI") OR EXCLUDE (SUBJAREA , "MEDI") OR EXCLUDE (SUBJAREA ,
"DECI") OR EXCLUDE (SUBJAREA , "PHYS") OR EXCLUDE (SUBJAREA ,

"BUSI") OR EXCLUDE (SUBJAREA , "ENER")) AND (EXCLUDE (SUBJAREA , "AGRI") OR EXCLUDE (SUBJAREA , "CENG") OR EXCLUDE (SUBJAREA , "EART") OR EXCLUDE (SUBJAREA , "PSYC") OR EXCLUDE (SUBJAREA , "BIOC") OR EXCLUDE (SUBJAREA , "MATE") OR EXCLUDE (SUBJAREA , "CHEM")) AND (EXCLUDE (SUBJAREA , "ENVI") OR EXCLUDE (SUBJAREA , "MULT") OR EXCLUDE (SUBJAREA , "NEUR") OR EXCLUDE (SUBJAREA , "Undefined")).

In order to assess the quality of the query, we checked that the authors we already knew to be relevant for our search (such as Jakob Nielsen, Donald Norman and Ben Shneiderman), appeared in the results in order to ensure that this search process was able to find these authors.

Reporting the SLR

The query resulted in a collection of 472 papers dating from 1982 to 2019. We did not put any restriction on the year of publication. Further inclusion and exclusion criteria were as follows:

Inclusion criteria:

- Papers that propose **functional** UI design principles.
- Papers that mention authors who propose functional UI design principles. This is needed to find relevant authors whose work is not appearing directly in the search, e.g., because it is not digitalized.

Exclusion criteria:

- Papers that describe the design of a system.
- Papers that state the need of the use for UI design principles, but do not reference principles and the authors who proposed them.

- Papers with a too narrow scope, e.g., describe principles for a specific style of interaction, e.g., touch based interaction, or specific type of users, e.g. blind and visually impaired users.
- Papers that refer to the term “design principles” in a different context than UI, e.g., in software engineering.
- Papers that describe other principles that are not referring to the functional aspects of an application, e.g. for layout design such as balance, rhythm, unity and harmony.

Based on these criteria, we retained 56 papers and discarded 416 papers.

To conclude the first step of the process, from these 56 papers, the most important authors need to be identified. Figure 2 shows the process followed for the selection of authors who propose functional UI design principles.

Figure 2. Process followed for the selection of authors.

From the selected papers we extracted a list of 41 authors who were cited in those 56 papers and who propose functional UI design principles. The list of authors is presented in Table 1 with the number of papers from the SLR that cite that particular author. The authors were cited in several contexts. Appendix D shows the complete list of contexts as presented in the 56 accepted papers.

Table 1. Authors who propose functional UI design principles.

Author	Mentioned by
Jakob Nielsen	32
Ben Shneiderman	29
Donald Norman	18
Catherine Plaisant	13
John Gould	5
Clayton Lewis	5
Rolf Molich	5
Larry Constantine	3
Lucy Lockwood	3
Jonathan Grudin	3
Jeff Johnson	2
Debbie Stone	2
Theo Mandel	2
Wilbert Galitz	1
Deborah Mayhew	1
Brad Mayer	1
Kevin Cox	1
David Walker.	1
Stephen Schach.	1
Jenny Preece	1
Harold Thimbleby	1
Deborah Hix	1
Rex Hartson	1
George Pangalos	1
Christopher Rivlin	1
Davies Cooper	1
William Lidwell	1
Kritina Holden	1
Jill Butler	1
Helen Sharp	1
Yvonne Rogers	1
Alan Dix	1
Janet Finlay	1
Gregory Abowd	1
Russell Beale	1
Eduard Tufte	1
Joseph Dumas	1
Janice Redish	1
Peter Polson	1
Aaron Marcus	1
Dan Olson	1

A number of authors were discarded because the principles they proposed were very domain or purpose specific and cannot be generalized. Examples are listed in Table 2.

Table 2. Examples of discarded authors for UI design principles with a narrow scope.

Author	Paper that mentions the author	Scope
Alma Whitten	Why Do Street-Smart People Do Stupid Things Online? (Bratus et al., 2008)	Principles for security specific challenges of matching system design to users' mental models
Doug Tygar		
Simson Garfinkel		
Marck Overmars	Usability heuristics in the context of control features on mobile games (Daud et al., 2016)	Principles for Iphone games
Robertas Damaševičius	Learning Object Reengineering Based On Principles for Usable User Interface Design (Damaševičius & Tankelevičienė, 2008)	Principles for mobile devices and e-learning
Lina Tankelevičienė		
Ka-Ping Yee.	N-Auth: Mobile authentication done right (Peeters et al., 2017)	Principles for secure systems

Given the long list of authors, and to ensure relevance of the found principles, we decided to analyze the authors who have been cited at least by 2 papers from the SLR and discard the rest of the authors. After analyzing the authors we also decided to discard Jonathan Grudin: he only presents the *consistency* design principle, which was previously proposed by other authors.

To further complete the set of authors of UI design principles, we also checked descriptions of UI design courses and books by doing a search in the entire web. These sources yielded four extra authors that were not found in the SLR: Wilfred Hansen (Hansen 1971), Brian Gaines, Mildred Shaw (Gaines & Shaw 1984), and Paul Heckel (Heckel 1991).

The final list of authors used to identify principles is as follows:

- (1) Ben Shneiderman
- (2) Jakob Nielsen
- (3) Donald Norman
- (4) Catherine Plaisant

- (5) John Gould
- (6) Clayton Lewis
- (7) Larry Constantine
- (8) Lucy Lockwood
- (9) Jeff Johnson
- (10) Debbie Stone
- (11) Rolf Molich
- (12) Theo Mandel
- (13) William Hansen
- (14) Paul Heckel
- (15) Brian Gaines
- (16) Mildred Shaw

2.2 *Identifying the principles per author*

Once we had the list of authors as sources for UI design principles, we compiled the principles proposed by each author. To that end, we performed two additional steps: 1) identification of the author's works where principles are proposed (note that some works are not immediately obtained from the SLR because they are, e.g., not digitalized) and 2) compilation of the principles proposed in those works. After having tried multiple strategies and based on the problems we experienced (see Section 5.1), the final strategy was as follows:

- (1) We selected the list of the most important authors as sources for principles from the SLR, descriptions of courses, books, etc. (see the previous section).
- (2) We used Harzing's Publish or Perish software as tool to retrieve and analyze academic citations using Google Scholar as data source. Publish or Perish is a software program that retrieves and analyzes academic citations. It uses Google

Scholar and Microsoft Academic Search to obtain the raw citations, then analyzes these and presents impact metrics. We look for the three most cited works of each author *that mention the principles (the original or rephrased principles)*.

- (3) Count the citation per author for those three works (note that if several authors proposed a principle together in a joint work the citation of that work is counted only once when counting citations for the principles).

Figure 3 shows the process followed for the identification of the most cited author's works in which they propose functional UI design principles. More details about the second step could be found in the section 5.1 with the specific strategy for finding the principles within the author's works.

Figure 3. Process followed for the selection of author's works.

Table 3 presents the resulting list of the most important authors ordered by the number of citations. "Cited by" counts the citation per author for the three most cited works where he/she proposed the principles. As not all authors have 3 works where they propose principles and some of them share works, the final list counts only 25 papers. "Average of citations" is "cited by" divided by the numbers of years since the paper was published.

Table 3. Most cited authors for UI principles and number of citations based on the three chosen papers by each author in Harzing's Publish or Perish.

Author	Cited by	Average of citations
Jakob Nielsen	22124	1005.64
Donald Norman	16703	596.54
Ben Shneiderman	13504	710.74
Rolf Molich	3607	133.59
John Gould	2871	92.61
Clayton Lewis	2657	85.71
Larry Constantine	1443	84.88
Lucy Lockwood	1253	73.71
Catherine Plaisant	966	54.21
Theo Mandel	467	21.23
Debbie Stone	442	40.18
Wilfred Hansen	345	7.66
Paul Heckel	230	7.19
Jeff Johnson	205	12.81
Brian Gaines	65	2.03
Mildred Shaw	65	2.03

After a careful review of the 3 most relevant works of the selected authors where they proposed the principles (see Table 3), we compiled all the principles proposed per author. The exercise of collecting the principles shows that: a) there is no single author who provides a complete set of principles and b) some of the sets proposed by different authors overlap. Table 4 shows the number of principles proposed by each author in their most cited works and the total number of principles (including variations: principles that slightly modifies the main ones) proposed in the three analyzed works per author. In total we collected 205 unique principles proposed in the most cited works, and 257 principles that include variations of principles (e.g., *feedback* and *offer informative feedback* are variations of the same principle) proposed by the same author in their two other analyzed works. The resulting long list of principles can be found in Appendix A. The next subsection provides a brief overview of the principles that made it to this long list.

Table 4. Number of principles proposed by the authors in their most cited work and variations.

Author	Number of principles in the most cited work	Number of principles including variations
Jakob Nielsen	10	20
Donald Norman	6	10
Ben Shneiderman	8	12
Catherine Plaisant ¹	8	8
Rolf Molich	9	16
John Gould	3	7
Clayton Lewis	3	11
Larry Constantine	6	6
Lucy Lockwood ²	6	6
Theo Mandel	24	27
Debbie Stone	8	8
Wilfred Hansen	14	17
Paul Heckel	30	30
Jeff Johnson	30	39
Brian Gaines	20	20
Mildred Shaw ³	20	20
Total	205	257

2.3 Summary of design principles proposed in the literature

In the early 70's (Hansen, 1971) defined a set of design principles for interactive graphics applications. Those principles were originally intended for clerical work with text editing applications, but the author claimed that they could be applied to administrative work as well. His first principle is *know the user* and gives advice

¹ Catherine Plaisant only proposed design principles in conjunction with Ben Shneiderman, although Ben Shneiderman also proposed principles without being co-author with Catherine Plaisant.

² Lucy Lockwood only proposed design principles in conjunction with Larry Constantine, although Larry Constantine also proposed the same principles in other later work without being co-author with Lucy Lockwood.

³ Mildred Shaw only proposed design principles in conjunction with Brian Gaines.

regarding which user information is important to gather, such as the user's education, experience, etc. Hansen explains that the user is a human, and therefore the designer should be aware of the user's capabilities and limitations. The rest of Hansen's principles aim to build a system according to those human capabilities and limitations. For example, the reason behind the principle *minimize memorization* when using the system is that humans have short-term memory limitations. Cockton (2004) categorizes Hansen's principles as "Directives for designers".

Norman (1983a) analyzed common errors made by users and derived design principles based on human cognition. He stresses how end-users often make errors and formulated guiding rules for avoiding those errors or at least avoiding negative impact of those errors.

Gaines and Shaw (1984) propose rules for dialogue engineering aiming at general principles for system design. However, these rules are quite high-level (for example: *We are responsible for computer behavior*) and therefore require interpretation.

Gould and Lewis (1985) present principles of system design, i.e., principles to be followed during the design process of a system. Gould and Lewis tested the principles and showed that not always are intuitive to designers.

Principles proposed by (Heckel, 1991) are another example of principles that unify aspects that can also be considered as advices for designers when driving the process of developing the UI (e.g. *avoid frustrating the user* and *make your design simple but not too simple*).

Nielsen and Molich (1990) performed a factor analysis of 249 usability problems and later (Nielsen, 1994c) proposed a refined set of ten heuristics. This set of heuristics is very well known in the field of UI design.

Shneiderman (1997) proposed eight principles that were derived from experience and have been refined over time. This author called those principles “Golden rules”. The golden rules, probably the most known principles in UI design, are general and require refinement for specific design domains but, as (Shneiderman, 2010) explains, the golden rules “have been well received as a useful guide to students and designers”. Those rules have also been expanded with variations by authors like (Johnson, 2007; Nielsen, 1995; Shneiderman & Plaisant, 2009).

Mandel (1997) proposes "three golden rules of UI design" and 24 principles that help to address those golden rules. This author is focused on placing users in control, reducing user's memory load and making the interface consistent.

Constantine and Lockwood (1999) present principles to design usable system with a user-centered design where the user is allowed to intervene during any stage of the process of obtaining the UI.

Evaluations against design heuristics or principles are proposed by (Stone et al., 2005). These authors group the error prevention and reversal of actions in a set of principles called *Tolerance*. They also emphasize the need for simplicity in the UI.

Besides proposing principles of what should be done in UI design, (Johnson, 2007) also provides a deep analysis on what can go wrong in UI design as affirmed by (MacKenzie, 2012). Johnson (2007) also takes into account the capabilities and limitations of the human cognitive system.

Although the previous mentioned principles were proposed many years ago, they continue to be used in the recent literature of the last few years. In the last decade renewed UI design principles have also been proposed. Shneiderman (2010) revisited the eight golden rules proposed by (Shneiderman, 1997) explaining their relevancy even for current types of UIs. Rather than proposing new principles, authors like (Johnson,

2007; Stone et al., 2005) compile the proposed principles and use them again, proving their permanent value in the UI design discipline.

Many of the analyzed sets of design principles are quite similar. There are principles that appear again and again, like consistency and error prevention. According to (Johnson, 2010) the rationale behind those design rules is based on human psychology.

3. Unifying the set of UI design principles (step three)

In order to answer the question “which are the core UI functional design principles”, first a reduction of the total number of principles (working from 257 principles resulting from the previous step) was made by removing overlaps and duplications.

Figure 4 shows the process we follow to unify the principles according to the different cases. This process shows the total number of principles resulting from each step.

Figure 4. Process followed for the unification of the principles.

To remove duplications, repeated and related principles were merged into a single principle. In this process, we identified different cases:

- (1) Principles that are proposed in joint work: the principles are proposed by several authors in a joint work and therefore appear as (pure) duplicates in the list. Examples of principles proposed by more than one author together in a joint work are: *Structure*, *Simplicity*, *Visibility*, *Feedback*, *Tolerance* and *Reuse*

proposed by Larry Constantine and Lucy Lockwood in (Constantine & Lockwood, 1999).

- (2) Principles that have a similar name: the name of a principle is a variation of the name of another principle. Examples of principles from different authors with variations in their names are: *Consistency* (Nielsen, 1994b), *Consistency and standards* (Nielsen, 1994a), *Consistency of the system* (Norman, 1983a), *Strive for consistency* (Shneiderman, 2010), *Be consistent* (Nielsen & Molich, 1990), *Intelligent consistency* (Lewis & Rieman, 1993), *Make the interface consistent* (Mandel, 1997), *Consistency* (Stone et al., 2005) and *Consistency, consistency, consistency* (Johnson, 2007). These nine principles were merged in the unified principle "*Strive for consistency*".
- (3) Principles that have a similar definition: the names of principles are not similar, but they do refer to the same concept. Examples of principles with different names but referring to the same concept are: *Access to system information* (Hansen, 1971): "With the access to the system information, the user does not need to remember what he said and is not kept in the dark about what is going on"; *Visibility of system status* (Nielsen, 1994c): "The system should always keep users informed about what is going on, through appropriate feedback within reasonable time".
- (4) Principles that are subsumed by others:
 - A specific principle is subsumed by a more general one: the principle is contained in another principle that is more general and that contains more information not implied by the specific one. For instance, *Names not numbers* (Hansen, 1971), *Reduce short-term memory load* (Shneiderman, 2010), *Promote*

an object-action syntax (intuitive) (Mandel, 1997) and *User progressive disclosure (context)* (Mandel, 1997), are all special cases of the more general principle (and thus subsumed by) *Minimize user's memory load* (Nielsen, 1994b).

- A principle that is split in two or more principles: The resulting split principles are then individually checked for subsumption, similar names, etc. For instance, *Provide immediate and reversible actions and feedback* (Mandel, 1997) is split in *Provide immediate and reversible actions* and *Provide feedback*. Then the first one is subsumed by *Actions should be reversible* (Norman, 1983a) and the second one is subsumed by *Offer informative feedback* (Shneiderman, 2010). So, in this case, neither the global or the split principles need to be kept.

Table 5 shows the number of principles that were finally obtained and how many of them were removed by each case. Out of the 257 variants of principles only 31 principles were unique and therefore not unified by the mentioned cases.

In order to minimize as much as possible the unification bias, the process was performed by the main author but continuously reviewed by the other two co-authors. The co-authors reviewed the lists of principles resulting of each step to ensure the consistency of the unification process along with the correct application of the selected strategy with the application of the 4 analyzed cases. Specifically the two final cases (subsumtion of principles by a general one and subsumtion of split principles) were double checked. In addition, disagreements found after this reviewing process were resolved by a discussion with co-authors. The final list of principles was reviewed by two senior researchers in UI design.

Table 5. Number of principles obtained by each case.

Case	Principles per case (out of total left)	Principles kept	Principles removed	Remaining Principles
Principles proposed in joint work	68 (out of 257)	34	34	223
Principles that have a similar name	118 (out of 223)	33	85	138
Principles that have a similar definition	48 (out of 138)	17	31	107
Principles subsumed by a more general one	49 (out of 107)	15	34	73
Principles split	2 (out of 73)	0	2	71

Further examples of principles reduction can be found in Appendix B. After this reduction process the set of unified principles comprised 71 principles.

As previously explained, many of the design principles are presented in similar ways by more than one author. Using this reduced set of 71 principles we made a matrix associating each of the principles with all the authors who propose it in some way (see Appendix C). For each pair of principle and author there are two possible values:

- 1 if the principle is proposed by the author, and
- 0 if the principle is not proposed by the author.

Using the Graph Commons social network analysis online software this binary matrix was represented as a network. Figure 5 shows the obtained network with the reduced set of design principles (red dots) and their relationship with their authors (black dots). This network has been made taking each author who proposed a principle individually into account, even when they worked together with other authors on the same set of principles. In this way, it is possible to know for each principle each author/s who proposed it, and for each author, all the principles that he/she proposed. In the figure, the size of the nodes is proportional to the number of connections the node has. The network can be accessed online through the following link:

<https://graphcommons.com/graphs/8f0b2e5e-264c-4ea5-9c00-5daf4a41fffc>

Figure 5. Network graph showing the relationships between design principles (red dots) and their authors (black dots).

As such, the network graph visualizes how principles are shared by the different authors. A measure of node centrality, namely “degree”, was used to assess the importance of individual nodes of principles. The degree is one of the most common measures of centrality and measures the number of ties that a node has. As explained by (Yan & Ding, 2009), in general, nodes with higher degree or more connections are more central to the structure and tend to have a greater capacity to influence others. In this specific work, a high degree is associated to the principles proposed in co-authored work or different authors who proposed similar principles in many papers, which can be interpreted as a principle considered important by relevant authors analyzed in this

paper. The size of the nodes in the network represents their degree: the more authors proposed a principle, the bigger the principle node; the more principles an author proposes, the bigger the author's node.

The number of vertices adjacent to a given vertex in a symmetric graph is the degree of that vertex. The normalized degree centrality is the degree divided by the maximum possible degree expressed as a percentage (Freeman, 1979). In this case, the centrality degree amounts to the proportion of authors citing this principle, with a range between 0 and 1. Out of the 71 unified principles, 50 are shared by at least 2 authors.

Table 6 shows the ranking of these 50 principles according to their centrality degree. This table is provided as a complementary information to the network graph previously presented. Both, the network graph and the centrality degree of authors, gives a general idea to the reader of how different authors have come up with similar design principles.

Table 6. 50 Design principles with the highest centrality degree.

Nr	Design principle	Centrality degree
1	Offer informative feedback	0.81
2	Strive for consistency	0.56
3	Simple and natural dialog	0.50
4	Know the user	0.50
5	Minimize user's memory load	0.50
6	Actions should be reversible	0.50
7	Prevent errors	0.50
8	Give the User Control	0.44
9	Make things visible	0.44
10	Structure the User's Interface	0.38
11	Iterative design to remove usability problems	0.31
12	Speak the User's Language	0.25
13	Provide shortcuts	0.25
14	Good error messages	0.25
15	Empirical measurement	0.25
16	Allow users to change focus (interruptible)	0.25
17	Provide visual cues (inform)	0.25
18	Use real-world metaphors (transfer)	0.25
19	Understand the tasks	0.25

Nr	Design principle	Centrality degree
20	Provide clearly marked exits	0.19
21	Help users recognize, diagnose and recover from errors	0.19
22	Provide a good conceptual model of the system	0.19
23	Accommodate users with different skill levels (accessible)	0.19
24	Recognition rather than recall	0.13
25	Flexibility and efficiency of use	0.13
26	Affordance	0.13
27	Involve users during the design	0.13
28	Integrated design	0.13
29	Reuse	0.13
30	Allow users to customize the interface (preferences)	0.13
31	Encourage exploration (predictable)	0.13
32	Display inertia	0.13
33	Lever the User's Knowledge	0.13
34	Avoid Frustrating the User	0.13
35	Respond to the User's Actions	0.13
36	We are all responsible for computer behavior	0.13
37	Computers provide a new medium for communication	0.13
38	Computers programs encode expertise	0.13
39	The style of dialog varies with the computer technology used	0.13
40	Be aware of the repertoire of techniques for person-computer dialog	0.13
41	Natural language may be used in a variety of different ways	0.13
42	Fluent language may not imply fluent understanding	0.13
43	Programs require presentation	0.13
44	Programs should be self-sufficient	0.13
45	Dialog should be logged and analyzed	0.13
46	Design for a changing and uncertain future	0.13
47	Make the best use of today's technology today and tomorrow's tomorrow	0.13
48	Cater to universal usability	0.13
49	Design dialogs to yield closure	0.13
50	Support internal locus of control	0.13

4. The most cited UI design principles (step four)

As 71 is still a too large number to manage easily, the set of principles needs further reduction to obtain a set that can be used as "core" set of guiding principles for junior UI designers.

We selected the core principles based on the number of citations per principle (and their variants). Citation count is an important factor to measure the influence of academic publications, where highly citation number indicates the recognition among

peers in a field of knowledge (Zhu & Ban, 2018). According to (Leydesdorff, 2008) “citation counts reflect the usefulness of the publication in the construction of new knowledge claims at other places in the network”. A high number of citations is an objective measure that has been wide spread and shows that a given paper has provoked an impact in the research community. An initial check was performed to verify if the authors whose works about design principles have been more cited are those that have more presence in courses about teaching UI design: The number of citations was positively connected with the authors who appear more often in the literature for UI education: the most cited authors are those presented as most relevant authors in the domain of UI design.

To count the citations of each of the 71 unique principles, we sum the citations of the works where each principle was proposed.

Figure 6 shows a chart with the design principles and their citation number. A few principles have a much higher citation than the rest, with a maximum of 59109 and a minimum of 37. The average number of citations is 7060.

Figure 6. Design principles and citation number.

Table 7 shows how many principles are cited more than a certain cutoff number: 19 principles have a very large citation number (i.e. higher than 10000). Between more than 1000 and 500 citations, there is not a big difference: there are only 5 principles more in the second category. Finally, we see that 63.78% of the principles are cited more than 200 times.

Table 7. Number of principles grouped by citation number.

Cutoff	Quantity	% of the total number of principles
Greater than 10,000 citations	19	26.76
Greater than 1,000 citations	31	43.66
Greater than 500 citations	36	50.70
Greater than 100 citations	45	63.38

Since our goal is to select the most important principles, and not the biggest number of principles, we decided to have a cutoff at 500. Although, the resulting set of principles only represents 50.70% of the total, the remainder of the principles represent a set of more or less equal size, but much less cited, as it can be seen on Figure 6. The citations of the 36 most cited principles also represent 98.98 % of the total citations of the 71 principles.

Table 8 shows the 36 most cited principles, ordered by citation number (Note that the numbers of the principles match with the numbers of Table 6 for an easier finding). The first five principles: *Offer informative feedback*, *Strive for consistency*, *Prevent errors*, *Minimize user's memory load* and *Simple and natural dialog* have the highest citation number and the highest number of authors that proposed the principles. However, a high number of authors that propose a principle does not necessarily imply that the work where the principle was proposed has a high citation number. For example, the principle *Provide clearly marked exits* was proposed by 3 authors and has

a citation number of 23438 as opposed to *Know the user* that was proposed by 8 authors and has a citation number 3661 (although still higher than 1000). In general, the principles having a citation number lower than 500 have been proposed by only one or two authors.

Clearly, this list of principles can be further structured along with a number of categories of principles, such as navigation, context of use, clarity, etc. There are many authors suggesting such classifications. As there seems to be no agreed-upon classification, the further structuring would require, e.g., collecting the opinions of several experts through e.g. a workshop or Delphi Panel. We consider this to be beyond the scope of this paper.

Table 8. 36 Design principles with the highest citation and year of the oldest of the highest cited works.

Nr	Principle	Number of authors that proposed it	Citation	First year it was proposed
1	Offer informative feedback	13	59109	1971
2	Strive for consistency	9	41868	1983
7	Prevent errors	8	39922	1971
5	Minimize user's memory load	8	37352	1971
3	Simple and natural dialog	8	28570	1990
14	Good error messages	4	23783	1971
13	Provide shortcuts	4	23708	1989
20	Provide clearly marked exits	3	23438	1989
12	Speak the user's language	4	23314	1984
18	Use real-world metaphors (transfer)	4	19818	1988
51	Help and documentation	1	19364	1994
9	Make things visible	7	19146	1988
26	Affordance	2	16691	1984
6	Actions should be reversible	8	16655	1971
22	Provide a good conceptual model of the system	3	16592	1988
49	Design dialogs to yield closure	2	13504	1984
48	Cater to universal usability	2	13234	1997
50	Support internal locus of control	2	13234	2004
52	Constraints	1	10040	2013
8	Give the user control	7	4109	1984
24	Recognition rather than recall	2	3776	1971
4	Know the user	8	3661	1971
21	Help users recognize, diagnose and recover from errors	3	3372	1990
25	Flexibility and efficiency of use	2	3309	1971
15	Empirical measurement	4	3113	1985
11	Iterative design to remove usability problems	5	2973	1984
10	Structure the user's interface	6	2897	1991
19	Understand the tasks	4	2322	1985
29	Reuse	2	1443	1999
16	Allow users to change focus	4	1204	1971
23	Accommodate users with different skill levels	3	1047	1991
28	Integrated design	2	892	1991
31	Encourage exploration	2	812	1971
30	Allow users to customize the interface	2	672	1997
17	Provide visual cues	4	670	1984
32	Display inertia	2	550	1971

5. Discussion

5.1 Lessons learned from the compilation of principles

While we compiled and analyzed the design principles we encountered many problems. Some of them are related to the search engine to identify works, others related to the identification of the principles inside the works.

Examples of pernicious problems related to using a search engine for the identification of the works where principles are proposed, are as follows:

- (1) A set of principles could be presented in more than one work of the author, so taking the citation of only one work is not enough. Google Scholar citation counts are not as reliable as Web of Science. When using Google Scholar we could see that is not possible to rank the works per citation. This search engine ranks sources using not only citation, but also uses relevance to rank publications. We could have used Web of Science or Scopus instead of Google Scholar but those citation counts are typically much lower as many resources are left out, while, as stated by (Harzing & Alakangas, 2016) the three databases provide sufficient stability of coverage to be used.
- (2) Sometimes, a principle that was proposed by an author does not appear in this search. Google Scholar seems to return incomplete results. One reason is that some works are too old and therefore not digitized, making the text not searchable by an engine. In those cases, the principle can only be found by human reading.
- (3) Sometimes an author has published many works, and the principles are not necessarily presented in the first 10 most cited works.
- (4) When there are too many results for a search in Google Scholar, the engine stops after 1000 results.

According to the strategy we used (see subsection 2.2) we analyzed the first three most cited works of each author *that mention the principles (the original or rephrased principles)*. This required going through the ranked publication list and checking in every work if it mentions principles. The rule we applied was as follows:

If an author has many papers and after reading 10 of them the principles cannot be found: Look for the works of the author where the words "principles", "rules" or "heuristics" are present in the body of the work. Figure 7 shows the steps followed to find the principles.

Figure 7. Strategy followed for finding the principles within the author's works.

The remaining problems of the strategy we performed were:

- (1) Sometimes an author has published many works and it requires going through many papers to find if he/she presents the principles or not. Since the list is reduced with respect to the previous strategy, we nevertheless managed to go through the full lists.
- (2) Sometimes it is not possible to check the works of an author (when the work only appears as a citation but not the work itself).

We also faced difficulties during the verification of the work of the analyzed authors. Sometimes it required a careful manual verification of each individual work for the following reasons:

- (1) Since a same principle may be phrased slightly differently in different works, an automated search would not be able to find all works where a principle occurs. Manually searching each document is a necessity.
- (2) Sometimes a principle is only a word, (or it contains) a word, that is too much used (like feedback) and then so many works appear that is really hard to extract the needed information.
- (3) When a principle is only a word it is necessary to check inside every work if it is proposed there as a principle, and if there are many resources is really hard again to do it. Sometimes only a citation appears and the work itself cannot be downloaded. In those cases it is not possible to check inside the work to verify it is really about the principle.
- (4) When not the exact phrase of the principle, but some or all of its words, are present in a work it is necessary to check inside every work if it is proposed there as a principle, and if there are many resources is really hard again to do it. Sometimes a citation only appears, so it is not possible to check inside.

While a manual search has the potential drawback of (besides being much slower) being less systematic and exhaustive than an automated search, in this case manually verifying each document was the better option. It allowed to find the different variants of each principle (variations by the same author or by other authors), and to identify duplicated principles.

5.2 Limitations

The limitations of this study are related to the limitations inherent to the search of relevant authors who proposed UI design functional principles, searching for relevant works of those authors and the identification of unique principles during the investigation of a huge amount and variety of principles.

Even though the systematic search process we used gives us some guarantee for completeness, we can never be completely sure that no other relevant author exist beyond the ones we evaluated. In order to mitigate this risk we considered all the elements that contribute to completeness: the use of two large databases, a sufficient broad set of keywords, and additional backward and forward snowballing to search for additional papers. The fact that we identified only 4 extra authors with the snowballing on top of the 41 authors already covered by the papers found in the SLR is an indication of reasonable completeness of our search.

In addition, we tried several strategies to further identify the relevant works of each of the selected authors and the different variations of principles proposed by them. As explained in the previous subsection, each of these strategies had their own limitations.

Also the process for the identification of unique UI design principles may present some limitations. However, the identification of the overlapping and duplication were done by the main author and reviewed by the two co-authors. Even though we ensured reliability in the search process as much as possible, we cannot be sure that other researchers would obtain exactly the same results as presented in this paper. This is caused by the fact that the unification process always includes human judgment, in which subjective factors cannot be completely ruled out. In order to minimize these effects, each step was double checked and all the disagreements found at the end of

each step were resolved by a discussion between the co-authors. The final set of principles were also assessed and revised by two senior researchers in UI design.

However, there is still room for a formal evaluation by design practitioners and professional designers in future research. A broader study could be performed in order to include graphics or screen layout principles which are important for usability and accessibility.

This study did not assess the empirical validity and applicability of the design principles. In fact, not many of the analyzed papers actually provided evidence for the validation of the principles they proposed. This points us to the issue of the need for empirical validation of the principles. Although this issue is outside the scope of this study, it is evident that they are worthy of further investigation.

6. Conclusion

The main contributions of this paper are a compilation of design principles and a selection of those that can be considered as the most relevant ones.

Following the guidelines proposed by Kitchenham and Charters in 2007, we performed a SLR to first identify the relevant authors in this domain. A total of 472 papers were retrieved, out of them, 56 papers were selected as relevant and examined in detail. From the examined papers, we made a study of the 16 most referenced authors. We analyzed the three most cited works of each author and extracted all the principles and their variants from these works, compiling a total of 257 UI design functional design principles and variants. Some principles were similar by name, by definition, or subsumed by other principles. Making use of unification and citation metrics, a final set of 36 core principles was identified. These principles are considered by the majority of the analyzed authors and have a number of citations higher than 500. As important examples within this final list, we can find the following UI design principles: *Offer*

informative feedback, Strive for consistency, Prevent errors, Minimize user's memory load and Simple and natural dialog. The compilation of principles provides educators and UI designers with insights in what should be considered as the most important (or agreed-upon) principles, and provides a clearer path to teach, evaluate, learn and improve the functional design of a UI.

7. References

- Adikari, S., McDonald, C., & Campbell, J. (2009). Little design up-front: a design science approach to integrating usability into agile requirements engineering. *International Conference on Human-Computer Interaction*, 549–558.
- Ahmed, S. S. (2010). “Unity in Diversity” among Design Principles. *Design Principles & Practice: An International Journal*, 4(4).
- Akiki, P. A., Bandara, A. K., & Yu, Y. (2015). Adaptive model-driven user interface development systems. *ACM Computing Surveys*, 47(1).
- Bratus, S., Masone, C., & Smith, S. W. (2008). Why do street-smart people do stupid things online? *IEEE Security & Privacy*, 6(3), 71–74.
- Chorianopoulos, K. (2008). User interface design principles for interactive television applications. *Intl. Journal of Human-Computer Interaction*, 24(6), 556–573.
- Cockton, G. (2004). Value-centred HCI. *Proceedings of the Third Nordic Conference on Human-Computer Interaction*, 149–160.
- Constantine, L. L., & Lockwood, L. A. D. (1999). *Software for use: a practical guide to the models and methods of usage-centered design*. Pearson Education.
- Damaševičius, R., & Tankelevičienė, L. (2008). Learning Object Reengineering Based On Principles for Usable User Interface Design. *Proc. of 10th Int. Conf. on Enterprise Information Systems, ICEIS*, 124–129.
- Daud, E. H. C., Mokhtar, S. A., & Mohd, F. (2016). Usability heuristics in the context of control features on mobile games. *2016 International Conference on Information and Communication Technology (ICICTM)*, 194–197.
- Estock, J. L., Li, A., Casey, M. C., Taylor, M. A., Boudreaux-Kelly, M. Y., Choma, M. D., Jacobs, K., & Eibling, D. E. (2018). Assessing Use Errors Related to the Interface Design of Electrosurgical Units. *AORN Journal*, 107(1), 72–82.
- Folmer, E., & Bosch, J. (2004). Architecting for usability: a survey. *Journal of Systems and Software*, 70(1–2), 61–78.
- Freeman, L. C. (1979). Centrality in Social Networks: Conceptual clarification. *Social Networks*, 1, 215–239.
- Gaines, B. R., & Shaw, M. L. G. (1984). *The art of computer conversation: a new medium for communication*. Prentice Hall International (UK) Ltd.
- Galitz, W. O. (2007). *The essential guide to user interface design: an introduction to GUI design principles and techniques*. John Wiley & Sons.
- Gould, J. D., & Lewis, C. (1985). Designing for usability: key principles and what designers think. *Communications of the ACM*, 28(3), 300–311.
- Hansen, W. J. (1971). User engineering principles for interactive systems. *Proceedings of the November 16-18, 1971, Fall Joint Computer Conference*, 523–532.
- Harzing, A.-W., & Alakangas, S. (2016). Google Scholar, Scopus and the Web of Science: a longitudinal and cross-disciplinary comparison. *Scientometrics*, 106(2), 787–804.
- Heckel, P. (1991). *The elements of friendly software design*. SYBEX Inc.
- Hentati, M., Ben Ammar, L., Trabelsi, A., & Mahfoudhi, A. (2016). A fuzzy-logic system for the user interface usability measurement. In *IEEE/ACIS (Ed.), 17th International*

- Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD* (pp. 133–138).
- Johnson, J. (2007). *GUI bloopers 2.0: common user interface design don'ts and dos*. Morgan Kaufmann.
- Johnson, J. (2010). *Designing with the mind in mind: Simple guide to understanding user interface design rules*. Morgan Kaufmann.
- Kimball, M. A. (2013). Visual design principles: an empirical study of design lore. *Journal of Technical Writing and Communication*, 43(1), 3–41.
- Kitchenham, B., & Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering. In *Engineering* (Vol. 2, Issue 1051).
- Lewis, C., & Rieman, J. (1993). *Task-center user interface design*.
- Leydesdorff, L. (2008). Caveats for the use of citation indicators in research and journal evaluations. *Journal of the American Society for Information Science and Technology*, 59(2), 278–287.
- MacKenzie, I. S. (2012). *Human-computer interaction: An empirical research perspective*. Newnes.
- Mandel, T. (1997). *The elements of user interface design* (Vol. 20). Wiley New York.
- Martins, L. C. G., & Garcia, R. E. (2015). Validation of User Interface Model: a Systematic Literature Review. *International Conference Software Engineering Research and Practice*, 145–151.
- Miller, K., Capan, M., Weldon, D., Noaiseh, Y., Kowalski, R., Kraft, R., Schwartz, S., Weintraub, W. S., & Arnold, R. (2018). The Design of Decisions: Matching clinical decision support recommendations to Nielsen's design heuristics. *International Journal of Medical Informatics*.
- Min, K., & Lee, S. (2019). Designing for “Raressential” Functions: Usage Patterns and UI Guidelines for Infrequent but Essential Tasks. *International Journal of Human–Computer Interaction*, 35(18), 1706–1728.
- Molina, A. I., Giraldo, W. J., Gallardo, J., Redondo, M. A., Ortega, M., & García, G. (2012). CIAT-GUI: A MDE-compliant environment for developing Graphical User Interfaces of information systems. *Advances in Engineering Software*, 52, 10–29.
- Myers, B., Hudson, S. E., & Pausch, R. (2000). Past, present, and future of user interface software tools. *ACM Transactions on Computer-Human Interaction*, 7(1), 3–28.
- Nielsen, J. (1994a). *Heuristic evaluation. Usability inspection methods* (J. W. & Sons (ed.)).
- Nielsen, J. (1994b). *Usability engineering*. Elsevier.
- Nielsen, J. (1995). *10 usability heuristics for user interface design*. Nielsen Norman Group.
- Nielsen, J. (1994c). Enhancing the explanatory power of usability heuristics. *Conference on Human Factors in Computing Systems*, 152–158.
- Nielsen, J., & Molich, R. (1990). Heuristic evaluation of user interfaces. *Conference on Human Factors in Computing Systems*, 249–256.
- Norman, D. A. (1983a). Design rules based on analyses of human error. *Communications of the ACM*, 26(4), 254–258.
- Norman, D. A. (1983b). Design principles for human-computer interfaces. *SIGCHI*, 1–10.
- Park, J. Y. (2009). A conceptual framework for a systematic mapping of layout design principles by using Yin and Yang theory. *Design Principles and Practices: An International Journal*, 3(1), 369–382.
- Peeters, R., Hermans, J., Maene, P., Grenman, K., Halunen, K., & Häikiö, J. (2017). n-auth: Mobile authentication done right. *Proceedings of the 33rd Annual Computer Security Applications Conference*, 1–15.
- Petrovčič, A., Taipale, S., Rogelj, A., & Dolničar, V. (2018). Design of mobile phones for older adults: An empirical analysis of design guidelines and checklists for feature phones and smartphones. *International Journal of Human–Computer Interaction*, 34(3), 251–264.
- Polack-Wahl, J. A. (2004). Teaching HCI in software engineering. *Frontiers in Education*, 2004. *FIE 2004. 34th Annual*, F1C–31–6 Vol. 2.
- Scapin, D. L., & Bastien, J. M. C. (1997). Ergonomic criteria for evaluating the ergonomic quality of interactive systems. *Behaviour & Information Technology*, 16(4–5), 220–231.

- Seffah, A., & Metzker, E. (2004). The obstacles and myths of usability and software engineering. *Communications of the ACM*, 47(12), 71–76.
- Shneiderman, B. (1997). *Designing the user interface: strategies for effective human-computer interaction* (5th ed., Vol. 3). Addison-Wesley Reading, MA.
- Shneiderman, B. (2010). *Designing the user interface: strategies for effective human-computer interaction* (5th ed., Vol. 3). Addison-Wesley.
- Shneiderman, B., & Plaisant, C. (2005). *Designing the user interface 4 th edition*. Ed: Pearson Addison Wesley, USA.
- Shneiderman, B., & Plaisant, C. (2009). *Designing the User Interface: Strategies for Effective Human-Computer Interaction* (5th ed.). Pearson Addison-Wesley.
- Smith, S. L., & Mosier, J. N. (1986). *Guidelines for designing user interface software*. Citeseer.
- Stone, D., Jarrett, C., Woodroffe, M., & Minocha, S. (2005). *User interface design and evaluation*. Morgan Kaufmann.
- van Schaik, P., & Ling, J. (2008). Modelling user experience with web sites: Usability, hedonic value, beauty and goodness. *Interacting with Computers*, 20(3), 419–432.
<http://dx.doi.org/10.1016/j.intcom.2008.03.001>
- Yan, E., & Ding, Y. (2009). Applying centrality measures to impact analysis: A coauthorship network analysis. *Journal of the American Society for Information Science and Technology*, 60(10), 2107–2118.
- Yigitbas, E., Stahl, H., Sauer, S., & Engels, G. (2017). Self-adaptive UIs: integrated model-driven development of UIs and their adaptations. *European Conference on Modelling Foundations and Applications*, 126–141.
- Zhu, X. P., & Ban, Z. (2018). Citation Count Prediction Based on Academic Network Features. *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*, 534–541.

Jenny Ruiz holds a PhD in Business Economics from the KU Leuven, Belgium. She is full professor in the Department of Informatics Engineering of the Faculty of Informatics and Mathematics, University of Holguin. Her research interests include User Interface design, model-driven engineering (MDE) and software engineering.

Estefanía Serral is a computer scientist currently working as an assistant professor at KU Leuven (Belgium). She has a highly international and interdisciplinary profile. Her research focuses on IoT, MDE and ubiquitous business processes. She has published more than 20 journal papers and many more refereed conference papers.

Monique Snoeck is full professor at the Research Center of Management Informatics at KU Leuven and visiting professor at the UNamur. Her research focuses on conceptual modelling, requirements engineering, software architecture, MDE, and e-learning technology. She has published over 30 journal papers and many more refereed conference papers.

Appendix A: authors and principles for UI design

Author	Principles
Jakob Nielsen	Simple and natural dialog
	Speak the User's Language
	Minimize user's memory load
	Consistency
	Provide feedback
	Provide clearly marked exits
	Provide shortcuts
	Good error messages
	Prevent errors
	Help and documentation
Donald Norman	Affordance
	Signifiers
	Constraints
	Mapping
	Feedback
	Provide a good conceptual model of the system
Ben Shneiderman (Also with Catherine Plaisant)	Strive for consistency
	Cater to universal usability
	Offer informative feedback
	Design dialogs to yield closure
	Prevent errors
	Permit easy reversal of actions
	Support internal locus of control
	Reduce short-term memory load
Rolf Molich	Simple and natural dialog
	Speak the User's Language
	Minimize user's memory load
	Be consistent
	Provide feedback
	Provide clearly marked exits
	Provide shortcuts
	Good error messages
	Error prevention
John Gould & Clayton Lewis	Early focus on users and tasks
	Empirical measurement
	Iterative design
Larry Constantine & Lucy Lookwood	Structure
	Simplicity
	Visibility
	Feedback
	Tolerance
Theo Mandel	Reuse
	Place users in control
	Use modes judiciously (modeless)
	Allow users to use either the keyboard or mouse (flexible)
	Allow users to change focus (interruptible)
	Display descriptive messages and text (Helpful)
	Provide immediate and reversible actions, and feedback (forgiving)

Author	Principles
	Provide meaningful paths and exits (navigable)
	Accommodate users with different skill levels (accessible)
	Make the user interface transparent (facilitative)
	Allow users to customize the interface (preferences)
	Allow users to directly manipulate interface objects (interactive)
	Reduce user's memory load
	Relieve short-term memory (remember)
	Rely on recognition, not recall (recognition)
	Provide visual cues (inform)
	Provide defaults, undo, and redo (forgiving)
	Provide interface shortcuts (frequency)
	Promote an object-action syntax (intuitive)
	Use real-world metaphors (transfer)
	User progressive disclosure (context)
	Promote visual clarity (organize)
	Make the interface consistent
	Sustain the context of user's task (continuity)
	Maintain consistency within and across products (experience)
	Keep interaction results the same (expectations)
	Provide aesthetic appeal and integrity (attitude)
	Encourage exploration (predictable)
Debbie Stone	Visibility
	Affordance
	Feedback
	Simplicity
	Structure
	Consistency
	Tolerance
	Accessibility
Wilfred Hansen	Know the user
	Minimize memorization
	Selection not entry
	Names not numbers
	Predictable behavior
	Access to system information
	Optimize operations
	Rapid execution of common operations
	Display inertia
	Muscle memory
	Reorganize command parameters
	Engineer for errors
	Good error messages
	Engineer out the common errors
	Reversible actions
	Redundancy
	Data structure integrity
Paul Heckel	Know Your Subject
	Know Your Audience
	Maintain the User's Interest

Author	Principles
	Communicate Visually
	Lever the User's Knowledge
	Speak the User's Language
	Communicate with Metaphors
	Focus the User's Attention
	Anticipate Problems in the User's Perception
	If You Can't Communicate It, Don't Do It
	Reduce the User's Defensiveness
	Give the User Control
	Support the Problem-Solving Process
	Avoid Frustrating the User
	Help the User Cope
	Respond to the User's Actions
	Don't Let the User Focus on Mechanics
	Help the User to Crystallize his Thoughts
	Involve the User
	Communicate in Specifics, Not Generalities
	Orient the User in the World
	Structure the User's Interface
	Make Your Product Reliable
	Serve Both the Novice and the Experienced User
	Develop and Maintain User Rapport
	Consider the First Impression
	Build a Model in the User's Mind
	Make Your Design Simple
	But Not Too Simple
	You Need Vision
Jeff Johnson	Focus on the user and their task, not on the technology
	Understand the user
	Understand the tasks
	Consider the context in which the software will function
	Consider function first, presentation later
	Develop a conceptual model
	Conform to the user's view of the task
	Strive for naturalness
	User user's vocabulary, not your own
	Keep program internals inside the program
	Find the correct point on the power/complexity tradeoff
	Design for the common case
	Make common results easy to achieve
	Two types of "common": how many users versus how often
	Design for core cases, don't sweat "edge" cases
	Don't complicate the user's task
	Don't give users extra problems
	Don't make users reason by elimination
	Facilitate learning
	Think outside-in, not inside-out
	Consistency, consistency, consistency
	Provide a low-risk environment

Author	Principles
	Deliver information, not just data
	Design displays carefully, get professional help
	The screen belongs to the user
	Preserve display inertia
	Design for responsiveness
	Acknowledge user actions instantly
	Let users know when software is busy and when it isn't
	Free users to do other things while waiting
	Animate movement smoothly and clearly
	Allow users to abort lengthy operations they don't want
	Allow users to estimate how much time operations will take
	Try to let users set their own work space
	Try it out on users, then fix it!
	Test results can surprise even experienced designers
	Schedule time to correct problems found by tests
	Testing has two goals: informational and social
	There are test for every time and purpose
Brian Gaines & Mildred Shaw	We are all responsible for computer behavior
	Computers provide a new medium for communication
	Computers programs encode expertise
	The user should be shown the choices available
	The system should adapts to the user under user control
	The system should minimize the user's workload by anticipating responses
	The system should cooperate with the user in validating responses
	It should be easy to escape from a dialog cleanly
	The style of dialog varies with the computer technology used
	Be aware of the repertoire of techniques for person-computer dialog
	Natural language may be used in a variety of different ways
	Fluent language may not imply fluent understanding
	Programs require presentation
	Know the market
	User should be informed
	Programs should be self-sufficient
	Design never ceases
	Dialog should be logged and analyzed
	Design for a changing and uncertain future
	Make the best use of today's technology today and tomorrow's tomorrow

Appendix B: examples of principles reduction

Principles proposed in joint work:

Principle	Proposed by
Structure	Constantine & Lockwood
Simplicity	
Visibility	
Feedback	
Tolerance	
Reuse	
Strive for consistency	Plaisant & Shneiderman
Cater to universal usability	
Offer informative feedback	
Design dialogs to yield closure	
Prevent errors	
Permit easy reversal of actions	
Support internal locus of control	
Reduce short-term memory load	
We are all responsible for computer behavior	Shaw & Gaines
Computers provide a new medium for communication	
Computers programs encode expertise	
The user should be shown the choices available	
The system should adapts to the user under user control	
The system should minimize the user's workload by anticipating responses	
The system should cooperate with the user in validating responses	
It should be easy to escape from a dialog cleanly	
The style of dialog varies with the computer technology used	
Be aware of the repertoire of techniques for person-computer dialog	
Natural language may be used in a variety of different ways	
Fluent language may not imply fluent understanding	
Programs require presentation	
Know the market	
User should be informed	
Programs should be self-sufficient	
Design never ceases	
Dialog should be logged and analyzed	
Design for a changing and uncertain future	
Make the best use of today's technology today and tomorrow's tomorrow	

Examples of principles that have a similar name:

Principle	Variations	Proposed by
Offer informative feedback	Provide feedback	Nielsen & Molich & Mandel
	Feedback	Norman & Lockwood, Constantine & Stone
	Offer informative feedback	Shneiderman & Plaisant
Strive for consistency	Consistency	Nielsen & Stone
	Consistency and standards	Nielsen
	Consistency of the system	Norman
	Strive for consistency	Shneiderman & Plaisant
	Be consistent	Molich
	Intelligent consistency	Lewis
	Make the interface consistent	Mandel
	Maintain consistency within and across products	Mandel
	Consistency, consistency, consistency	Johnson
Minimize user's memory load	Minimize user's memory load	Nielsen & Molich
	Reduce user's memory load	Mandel
	Minimize memorization (general)	Hansen
	The system should minimize the user's workload by anticipating responses	Gaines
Prevent errors	Prevent errors	Nielsen & Molich, Shneiderman & Plaisant
	Error prevention	Nielsen & Molich
	Engineer for errors	Hansen
	Engineer out of the common errors	Hansen
Simple and natural dialog	Simple and natural dialog	Nielsen & Molich
Speak the user's language	Speak the user's language	Nielsen & Molich & Heckel
	Use user's vocabulary, not your own	Johnson
Actions should be reversible	Actions should be reversible	Norman
	Permit easy reversal of actions	Shneiderman
	Reversible actions	Hansen
Provide shortcuts	Provide shortcuts	Nielsen ,& Molich
	Provide shortcuts for skilled users	Shneiderman
	Provide interface shortcuts (frequency)	Mandel
Good error messages	Good error messages	Nielsen & Molich & Hansen
	Precise and constructive error messages	Nielsen
	Provide good error messages	Molich
	Display descriptive messages and text (Helpful)	Mandel

Examples of principles that have a similar definition:

Principle	Variations	Proposed by
Provide feedback	Provide feedback	Nielsen & Molich & Mandel
	User should be informed	Gaines
Prevent errors	Prevent errors	Nielsen & Molich, Shneiderman & Plaisant
	The system should cooperate with the user in validating responses	Gaines
	Make Your Product Reliable	Heckel
	Provide a low-risk environment	Johnson
Visibility of system status	Visibility of system status	Nielsen
	Access to system information	Hansen
	Let users know when software is busy and when it isn't	Johnson
Help users recognize, diagnose and recover from errors	Help users recognize, diagnose and recover from errors	Nielsen
	Help the User Cope	Heckel
Actions should be reversible	Actions should be reversible	Norman
	Tolerance	Constantine & Lockwood & Stone
	Provide defaults, undo, and redo (forgiving)	Mandel
Empirical measurement	Empirical measurement	Molich & Gould & Lewis
	Try it out on users, then fix it!	Johnson
	Early and continual user testing	Gould & Lewis
Allow users to change focus (interruptible)	Allow users to change focus (interruptible)	Mandel
	Free users to do other things while waiting	Johnson
	Use modes judiciously (modeless)	Mandel
	It should be easy to escape from a dialog cleanly	Gaines
Allow users to customize the interface (preferences)	Allow users to customize the interface (preferences)	Mandel
	Try to let users set their own work space	Johnson
Provide visual cues (inform)	Provide visual cues (inform)	Mandel
	Orient the User in the World	Heckel
	The user should be shown the choices available	Gaines
Use real-world metaphors (transfer)	Use real-world metaphors (transfer)	Mandel
	Mapping	Norman
	Match between system and the real world	Nielsen

Examples of principles subsumed by a more general one:

General principle	Subsumed variations	Proposed by
-------------------	---------------------	-------------

Prevent errors	Prevent errors	Nielsen & Molich, Shneiderman & Plaisant
	Data structure integrity	Hansen
	Redundancy	Hansen
User control and freedom	User control and freedom	Nielsen
	Allow users to directly manipulate interface objects (interactive)	Mandel
Flexibility and efficiency of use	Flexibility and efficiency of use	Nielsen
	Optimize operations	Hansen
	Rapid execution of common operations	Hansen
	Muscle memory	Hansen
	Reorganize command parameters	Hansen
Empirical measurement	Empirical measurement	Molich & Gould & Lewis
	Test results can surprise even experienced designers	Johnson
	Schedule time to correct problems found by tests	Johnson
	Testing has two goals: informational and social	Johnson
	There are test for every time and purpose	Johnson
Early focus on users and tasks	Early focus on users and tasks	Gould & Lewis
	Consider the context in which the software will function	Johnson
Maintain the User's Interest	Maintain the User's Interest	Heckel
	Focus the User's Attention	Heckel
Lever the User's Knowledge	Lever the User's Knowledge	Heckel
	Think outside-in, not inside-out	Johnson
Avoid Frustrating the User	Avoid Frustrating the User	Heckel
	Don't complicate the user's task	Johnson
	Don't make users reason by elimination	Johnson
Design for the common case	Design for the common case	Johnson
	Make common results easy to achieve	Johnson
	Two types of "common": how many users versus how often	Johnson
	Design for core cases, don't sweat "edge" cases	Johnson

Principles subsumed by more than one principle:

Principle	Subsumed by	Proposed by
Provide immediate and reversible actions, and feedback (forgiving) (Mandel)	Provide feedback	Nielsen & Molich & Mandel
	Actions should be reversible	Norman
Early focus on users and tasks (Gould, Lewis)	Know the user	Molich & Hansen
	Understand the tasks	Johnson

Appendix C: principles for UI design and the authors who use them

	Jakob Nielsen	Donald Norman	Ben Shneiderman	Rolf Molich	John Gould	Clayton Lewis	Larry Constantine	Lucy Lockwood	Catherine Plaisant	Theo Mandel	William Hansen	Debbie Stone	Paul Heckel	Jeff Johnson	Brian Gaines	Mildred Shaw
Offer informative feedback	1	1	1	1			1	1	1	1	1	1		1	1	1
Strive for consistency	1	1	1	1		1			1	1		1		1		
Simple and natural dialog	1			1		1	1	1				1	1	1		
Know the user				1	1	1					1		1	1	1	1
Minimize user's memory load	1		1	1					1	1	1				1	1
Actions should be reversible		1	1				1	1	1	1	1	1				
Prevent errors	1		1	1					1		1		1		1	1
Give the user control	1		1							1			1	1	1	1
Make things visible		1				1	1	1		1		1	1			
Structure the user's interface						1	1	1		1		1	1			
Iterative design to remove usability problems				1	1	1									1	1
Speak the user's Language	1			1									1	1		
Provide shortcuts	1		1	1						1						
Good error messages	1			1						1	1					
Empirical measurement				1	1	1								1		
Allow users to change focus (interruptible)										1				1	1	1
Provide visual cues (inform)										1			1		1	1
Use real-world metaphors (transfer)	1	1								1			1			
Understand the tasks					1	1							1	1		
Provide clearly marked exits	1			1						1						
Help users recognize, diagnose and recover from errors	1		1										1			
Provide a good conceptual model of the system		1											1	1		
Accommodate users with different skill levels (accessible)										1		1	1			
Recognition rather than recall	1									1						
Flexibility and efficiency of use	1										1					
Affordance		1										1				
Involve users during the design				1									1			

[illegible]

[illegible]

Appendix D: context in which principles were cited

Context	Amount of papers
Design in general	12
Mobile application	10
Usability evaluation	8
Education	4
Security application	3
Application for impaired users	2
Access system	2
Games	2
Television system	2
Managing display layouts	2
E-commerce system	2
Medical application	2
Bioinformatics	1
Control rooms system	1
Workflow system	1
Museum system	1
Internet booking service	1
Ambient computer system	1
Operating system	1
System for children	1
System for carpoolers in the workplace	1
Photo library system	1
Web application	1
Computer agents	1
Bearing design support system	1
Panel-oriented interface	1
Adaptive user interfaces	1
Augmented reality	1
Managing location privacy	1
Hand speech system	1