GDI as an alternative guiding interaction style for Occasional users

Antonio L. Carrillo-León (carrillo@uma.es), Juan A. Falgueras-Cano
University of Málaga, Department of Computer Sciences, Teatinos, 29071 Málaga, Spain.

Abstract. It is usually taken for granted that *Direct Manipulation* is the best interaction style for inexperienced or non-expert users; moreover, this style of interaction is generally considered the best for almost every situation and user. The recent shifts in technology that we all are currently experiencing have given rise to a great deal of new kinds of users performing specific tasks in a variety of scenarios. In this paper, we focus on users who access a system occasionally, infrequently, or in an unplanned way; i.e., users who do not want or cannot afford a learning curve. We show that for them, *Direct Manipulation* is not always the most suitable style of interaction. We assess the advantages of guiding this kind of users, in particular through the guided interaction framework known as *Goal Driven Interaction*. GDI can be viewed as a superset of wizards providing support far beyond a few steps through dialogs. Indeed, GDI is an interaction style with characteristics of its own. We report a complete user test that backs up previous hypotheses. The analysis of empirical data proves that *GDI* is more time-efficient than *DM, requiring* fewer moderator assistances for the users. Post-test questionnaires confirmed that participants had a strong preference for *GDI*.

Keywords: Interaction design process and methods, Interaction paradigms, Empirical studies in interaction design, Interfaces for occasional users.

1 Introduction

Interaction Style is a classic topic discussed in the mainstream HCI literature, e.g. [43, 14]. There are two main general approaches to interact with computers: the conversational world and the model world, corresponding to sequential and freely accessible dialogues, respectively. The former includes Question and Answer dialogues, Command Line interfaces, Menu-based interaction, Form Fill-in data entry, Natural Language, etcetera; text is therefore the primary tool for interaction between the user and the system in this model. The other general approach, the model world, makes use of graphics and metaphors [8] as well as tools like Windows, Icons, Menus and Pointer (WIMP), to assist the user with asynchronous and free management of objects on the screen. The Direct Manipulation (DM) [42, 43, 18] of interactive elements makes this general approach even more familiar and intuitive. Ideally, the users see and predict the behavior of well-known objects through metaphors. They then follow their natural intuition to manipulate them, receiving immediate feedback: “You don’t notice the computer because you think of yourself as doing the task, not as using the computer”, Norman [35]. However, DM-WIMP interfaces still require time to become acquainted with them. In particular, Stasko stated: “...Although GUI and WIMP interfaces are a big step past line-oriented terminals, they still have a learning curve and they can be awkward to use”.

Whatever the case may be, the success of the DM approach has eclipsed the others. These interfaces are typically assumed to be the best solution for every possible scenario, task and user. However, the massive expansion of technology has caused the appearance of new scenarios of use in addition to quite different tasks and users for them. This requires revisiting the most appropriate approach for each case. The success of the interaction depends basically upon its suitability for the final user. There are many differences between how experts and novices perceive and use software applications [13]. On the one hand, experts have the best possible mental model thanks to their experience. As opposed to novices, they require less guidance and help. On the other hand, novices are users with little or no knowledge about the system. They initially have a deficient understanding about the possibilities of the application and have to focus on how to deal with the interface [28]. To gain expertise they follow a learning curve that is usually traversed through study and repeated use of the application. Moran [28] argued that for novices learning to use the interface is more important than being able to accomplish the task: “Learning is, of course, paramount for the novice whereas the time it takes to do a task is secondary”. Interestingly, there are situations where learnability is not as important as the time elapsed in the interaction. For example, purchasing a train ticket in a self-service machine when the train is on the verge of departing. The aim here is to accomplish the transaction as quickly as possible. In such cases, learning how to use the application is not as relevant as just getting it done. Achieving the goal, i.e., getting the ticket in your hand, becomes the priority, while learning during the interaction – an action that may not even be repeated at all in the future − becomes secondary (“[the user] does not have any ambition to master the system and may prefer to be led by the hand to accomplish…”) [46]. Then, some problems arise when a system is occasionally accessed and the user does not want or cannot afford a learning curve.

At least three arguments support an explicit design of systems for these occasional users: advances in UIs, new scenarios of use, and a more ambitious inclusive design. Firstly, the evolution of Graphical User Interface (GUI) [43, 30] and the correct inclusion of multi-touch systems [36, 40, 5] have brought about new kinds of devices and new ways of interaction. Secondly, the ubiquity and the permanent on-line interconnection of mobile devices have burst into unexplored contexts of use, e.g. shopping centres, leisure facilities, museums, airports… [11, 15, 41]. These technologies have facilitated new scenarios of use where human-computer interaction (HCI) is on the move, using technology as a mean to accomplish an immediate goal. Thirdly, this increase in the heterogeneity and number of occasional users is also driven by the incorporation of the principles of Accessibility [1, 29, 47], Usability [33, 32] and Inclusive Design [12, 39].

In summary, there is a wider spectrum of users able to perform not only planned but also occasional activities. This new scenario demands simplicity, immediacy and no previous knowledge, just like when buying a transport ticket in a self-service machine, visiting a new city, designing or furnishing a room, or using an audio-guide mobile app during the visit to a museum or archaeological site. Our aim is to check with real users which kind of interface − either guided or non-guided − they prefer. The selected guided interaction style, namely Goal Driven Interaction (GDI), is introduced in section 2. We compare a GDI-guided standard application for furnishing kitchens with its non-guided counterpart. In section 3 we present our experiment, and in section 4 we discuss the results.

2 Goals Driven Interaction

Goals Driven Interaction (GDI) [9] is conceived as a human-computer interaction style especially suitable for occasional users, that is, users who do not want or cannot afford a learning curve. This means that even if the user dealt previously with the same or analogous technology, there are a number of factors that make it unwise to rely on the user’s memory recall or implicit visual recognition as the main mechanisms to learn how to use the interface. Some of those factors could be the time elapsed since the last interaction, the difficulties users experience on learning, and, in many cases, their lack of interest [17]. It is recommended to assume that the user will have to cope with an unknown interface, with no requirement upon previous knowledge to use it. The corresponding style of interaction will be of conversational and sequential nature, even sacrificing the possibility of performing tasks in parallel as well as other typical advantages of pure DM interfaces. This does not prevent the user from deciding among alternatives. The objective here is to guide the user, step by step, in a hierarchical way along the whole interaction. Thus, hierarchically organized objectives and sub-objectives will be clearly presented to the user one at a time. The goals, once achieved, are left behind. Nevertheless, the user will have the possibility to return to past goals, via cancellation, whenever it is possible to do so. The user will be strictly guided on the "what to do" and the "how to do it," one step at a time. This strategy constrains the freedom of the DM in a similar way to processes guided by wizards [45, 27, 4, 48]. In some sense, GDI could be viewed as a layer over pure DM, making it a kind of assisted interaction, adding control over DM: it allows the user to touch, move or briefly interact with graphic objects, but only when the guide allows for it. It could also be viewed as a super-set of wizards. Standard DM transfers the control to the user whereas GDI guides them.

The fundamentals of GDI trace back to the work of [31] devoted to the mechanisms of human reasoning for problems resolution. Their vision (as in GDI) was based on breaking up the main or general goal into a hierarchical tree of sub-goals. The branches of that tree can have different lengths depending on the degree of fragmentation in sub-goals. The leaves of the tree are elementary actions or final goals that do not require further explanations nor decomposition. Based on this work, Card, Moran and Newell [6, 7] developed one of the most important cognitive models, the Human Processing Model, whose initial paradigm consisted in conceiving the interaction as a problem resolution task. They described a psychological model of humans comprising three interactive systems: perceptive, motor and cognitive. Each one would have its own memory and its own processor. This vision of the user as an information processing system permits the description of all the activities (both physical and mental) that take part in a task. This in turn gave origin to a family of methods for modelling, specifying and evaluating a user interface: the GOMS models [20-22]. A GOMS (Goals, Operators, Methods, and Selection rules) model is a description of the knowledge a user must have in order to carry out tasks and thereby goals on a device; it is a representation of the "how to do it". The aim of GDI is to preclude the user from having to devote time to acquire that knowledge. The user will not need to look anything up in any user manual, or help system, nor think about anything except navigating through the proposed steps. We only expect from the user to know the goal to achieve.

There are many versions derived from the original GOMS. In particular, Kieras [24, 25] contributed with the idea of a structured natural language, namely NGOMSL (Natural GOMS Language). These methodologies themselves are framed in a larger set of techniques enabling a hierarchical task analysis. The main goal of these techniques is the decomposition of a complex task in sub-tasks, so that the resolution method can be followed step by step. For instance, HTA [2], TAG [38], TKS [23], or the more recent CTT [37], that also includes the possibility of expressing the concurrency of the tasks. None of them were oriented to readiness but to formalization and have been discarded for our methodology. In contrast, NGOMSL presents a wide and detailed literature, as well as practical construction methods, e.g. [24, 25], that can be broadly applied. But above all, the most interesting feature of GDI is their closeness to the user's natural language: any user can read and follow it, as a recipe. NGOMSL can be used to specify GDI interfaces but it could not qualify to model some situations and aspects of GDI. To overcome this issue the authors have extended and adapted NGOMSL, for instance by including the possibility of cancelling goals. This extended version can be used as a source specification language [9].

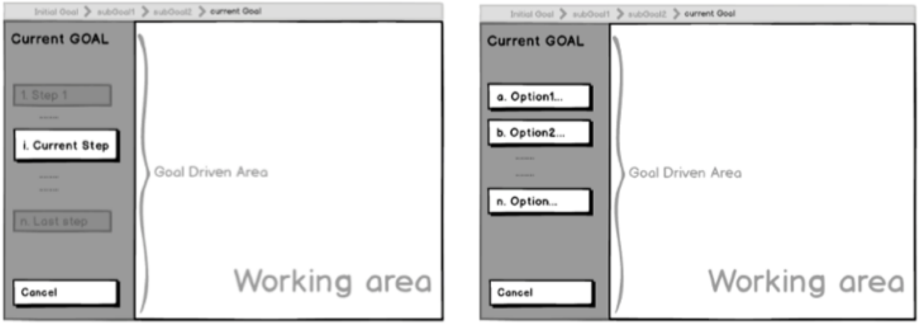


Fig. 1. Proposed layout of a *GDI interface*, when offering a *Method* (on the left) or a *Selection* (on the right)

From the specification obtained applying our extension of NGOMSL is possible, after an easy compilation process, to obtain the corresponding guide of the final GDI interface, which composes the Goal Driven Area. GDA is what explicitly guides the user through the hierarchy of goals (Fig. 1). This area becomes the substitute for the typical menus and toolbars used in standard MD-WIMP, not necessary for these guided interfaces. GDA will always show the method or the selection that allows the user to accomplish the current goal. If this goal requires the user to make a selection, GDA shows the different excluding options (alternatives) that compose that selection, so that the user can choose one of them. In case the goal must be accomplished following a specific method, GDA will offer the sequence of steps that compose this method. The interface underlines the current step, which could imply the initiation of a new subgoal (and then another method or selection), or the realization of an elementary action (without methods nor selections).

3 GDI vs Standard DM Interfaces: a Comparative Study

At the beginning of the previous section, we explained that GDI was proposed as a way to help occasional users who do not want or cannot afford a learning curve for specific goals. Our aim has been to test this postulate with real users. To that end, a comparative study between guided and non-guided interaction has been conducted. This empirical analysis gave us objective measurements and subjective opinions. We wanted to test 2 objectively measurable hypothesis, related to time and errors, and a third subjective hypothesis, related to satisfaction.

For non-computer experts, occasional users:

- (H1) they proceed quicker with GDI than with standard DM,
- (H2) they make fewer errors with GDI than with standard DM,
- (H3) they prefer GDI to standard DM.

As can be seen, the 3 pillars of usability are covered: efficiency (H1), effectivity (H2), and satisfaction (H3).

A first approach to this study was presented in [10] as "work in progress". However, that work could only show what appeared to be a trend, since it had very few participants, and lacked analysis, discussion and statistical study about the significance of the results.

3.1 Participants

The study involved twenty volunteer participants (n=20), a number that meets the criteria of [26, 34, 44, 3]. There were 12 females and 8 males. Their ages ranged from 10 to 52 years, with a mean age of 44 years. All subjects were unaware of the final aim of the research. They had not participated in previous usability studies nor had they received any incentives to participate in the experiment. Each participant had to use two different versions of the same application (the two types of interfaces to be compared), following a *counterbalanced* strategy (as we will explain later).

Table 1 summarizes the main data and characteristics of the participants, collected by means of a Background Questionnaire.

To do the recruiting and screening of the participants, the primary requirement was no previous experience with similar applications to those tested in this study. Only two exceptions were intentionally included: two participants were professionals at the task domain. Both of them had this task as part of their daily activity using similar commercial direct manipulation software packages. The other screening criterion was to discard computer expert users, as reflected in Table 1: no participant had a self-graded Computer Expertise greater than 3 on a 5–point scale. This parameter is subjective. To confirm these self-assigned values, we also asked several complementary questions: PUS, KDSK, TSE, TDK, ASL, and ORK (see Table1). A complementary study with Eye Tracker was conducted for two of the participants. See section 3.6.4 for details.

Table 1. Overview of the main data and characteristics of the participants, collected by means of the *Background Questionnaire*: *Gender* (**G**), *Age* (**A**), ***Computer Expertise*** (**CE**), *Previously Used Software* (**PUS**) [options: **I**nternet, **E**mail, **W**ord Processor, **S**pread_sheet, **M**ultimedia, **A**ccounting & management, **D**atabases, social **N**etworks, **O**thers], *Kitchen Design Software Knowledge* (**KDSK**), *Tablets and Smartphones Expertise* (**TSE**), *Task Domain Knowledge* (**TDK**), *Academic Studies Level* (**ASL**) ranged from 0 to 3 [0:No studies; 1:Primary school; 2:Secondary school; 3:University], and *Other Relevant Knowledge* (**ORK**). The *Computer Expertise* (**CE**) of each participant was checked and validated in accordance with these items.

#	CE	PUS	KDSK	TSE	TDK	ASL	ORK
	[0..5]	[I,E,W,S,M,A,D,N,O]		[0..4]	[0..4]	[0..3]	
1	0		No	1	3	1	
2	0		No	1	3	1	
3	0	I	No	1	3	1	
4	0	I	No	2	1	0	
5	1	I,E	No	2	3	1	
6	1	I,M	No	2	3	2	
7	1	I,M,O	No	2	2	1	Occasional use of IKEA kitchen design app
8	2	I,E,W,S,M,A (forgotten)	No	2	3	2	Forgotten computer courses
9	2	I,E,W,M,N,O	No	2	3	1	
10	2	I,E,W,M,O	No	2	3	1	Medical software
11	2	I,E,W,M,O	No	2	3	3	Medical software
12	2	I,E,W,S,M,O (forgotten)	No	2	3	1	Pantograph. Forgotten computer courses
13	2	I,E,W,M,A,N,O	No	2	2	1	Photoshop
14	3	I,E,W,S,M,A,D	No	2	3	2	
15	3	I,E,W,S,M,A,D,	No	2	2	3	Veterinary software
16	3	I,E,W,M,O	No	2	3	1	AutoCAD and Presto user
17	3	I,E,W,S,M,A,D,	No	1	3	2	Forgotten computer courses

18	3	I,E,W,S,M,A,D,	No	2	3	2	
19	3	I,E,W,S,M,A,O	kitchens	3	4	2	Photoshop. Kitchen design professional
20	3	I,E,W,S,M,O	kitchens	2	4	1	Kitchen design professional

3.2 Materials and Tasks

Although other applications could be plausible, we chose for our study a familiar domain, i.e. kitchen furniture. This scenario comprises three development stages: designing, furnishing and final adjustments of a kitchen. Some of the involved tasks require extensive object manipulation, usually implemented with standard DM interfaces. These characteristics could lead us to think that this is an application especially suitable for a classic DM interaction style, making it particularly challenging for our study.

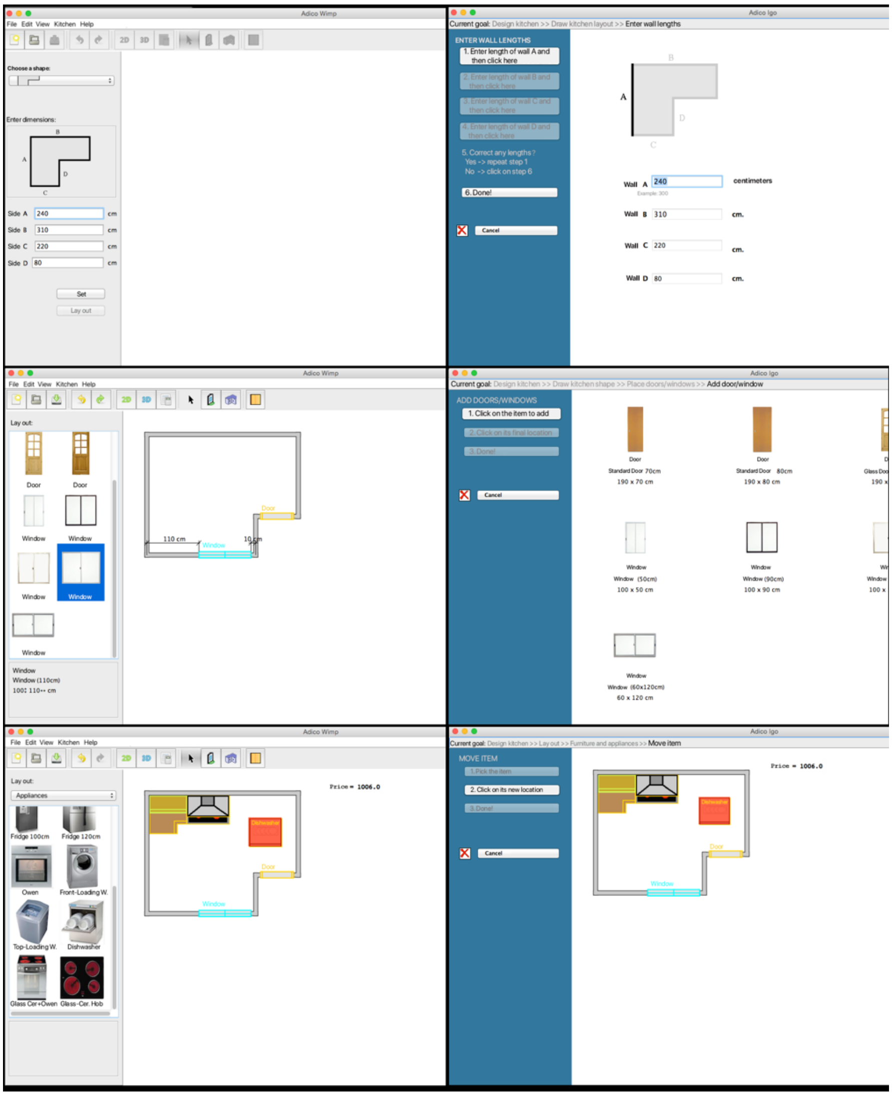
We developed two versions of the same application (in Java language), one with a GDI interface and the other with a classic DM interface (Figure 2). Both share most of the code except for the sections in which the user interfaces are involved. Both offer the same functionality and allow the user to achieve the same goals. The DM version is inspired by the desktop version offered by IKEA [19]. Figure 2 depicts a sub-task per row for each of the stages the user goes through. The left column corresponds to the DM interface whereas the right column is related to the GDI interface. The DM interface is a plain standard WIMP application. The user first introduces the shape and the dimensions of the kitchen and then furnishes it. To this end, the user has to pick elements and position them inside the boundaries following the canonical DM way, i.e. with undo/re-do, tool-tips for buttons, right-click pop-up menus, etc.

Regarding the three screenshots on the right column for the GDI interface, note that the user is presented with a list of actions to be read and followed. Each action is atomically performed by short DM interactions in the Working Area. The user cannot continue until the previous action is completed, as we explained in Section 2.

A screencast of a participant using GDI can be watched in https://youtu.be/4owtCVMuqc4

A screencast of a participant using DM is shown in https://youtu.be/QeafXLlsXzY

The source code for both applications is available in https://github.com/juanfal/ADICO



Current goal: Design kitchen >> Draw kitchen layout >> Enter wall lengths

ENTER WALL LENGTHS

1. Enter length of wall A and then click here

2. Enter length of wall B and then click here

3. Enter length of wall C and then click here

4. Enter length of wall D and then click here

5. Correct any lengths?
Yes -> repeat step 1
No -> click on step 6

6. Done!

Wall A 240 centimeters
Example: 300

Wall B 310 cm.

Wall C 220 cm.

Wall D 80 cm.

Choose a shape:

Enter dimensions:

Side A 240 cm

Side B 310 cm

Side C 220 cm

Side D 80 cm

Current goal: Design kitchen >> Draw kitchen shape >> Place doors/windows >> Add door/window

ADD DOORS/WINDOWS

1. Click on the item to add

2. Click on its final location

3. Done!

Door
Standard Door 70cm
190 x 70 cm

Door
Standard Door 80cm
190 x 80 cm

Window
Window (50cm)
100 x 50 cm

Window
Window (90cm)
100 x 90 cm

Window
Window (60x120cm)
60 x 120 cm

Window
Window (110cm)
100: 110↔ cm

Current goal: Design kitchen >> Lay out >> Furniture and appliances >> Move item

MOVE ITEM

1. Pick the item

2. Click on its new location

3. Done!

Price = 1006.0

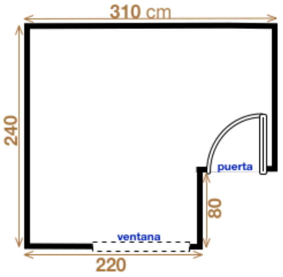
Fig. 2. Screenshots of the tested interfaces: a) On the left, the DM interface; b) On the right, the development of the same task using the GDI interface. The first pair of screenshots is related to the process of entering the dimensions of the kitchen walls. In the second pair the user is selecting and adding a window. In the third pair, the user is moving furniture around.

Each participant was given a single piece of paper containing the information shown below. It briefly described the application scenario along with the three groups of tasks chosen for the test:

Imagine yourself arriving at a kitchen furniture shop. All employees are busy. However, a computer is available for you to specify your preferences for your kitchen: shape, dimensions, the furniture you want to buy and where you would particularly like to place it, make further adjustments, etc. To save time, you decide to use this system.

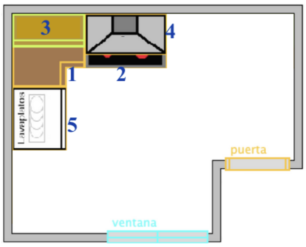
Task 1: Specifying how your kitchen is
Suppose that you bring a sketch (as the figure below) with the shape of your kitchen, the dimensions of the walls, doors and windows, and the location of all of them:

- Door 190 (high) x 70 (width) cm
- Window 100 (high) x 110 (width) cm



Task 2: Choosing furniture
Suppose the furniture you want to buy is listed below.
Choose and place the pieces as shown in the figure below:

1- Kitchen corner (80 cm wide)
2- Hob with Oven included
3- Showcase (80 cm, with 2 doors)
4- Metal extractor hood
5- Dishwasher
Write down the budget:
___(euros)



Task 3: Retouching and ordering

- Place the dishwasher next to the hob.
- A family is going to give you a metal extractor hood, therefore remove ours.
- Observe the kitchen in 3D.
- Look and see the detailed kitchen budget and note down the number of chosen items: ____ and the final budget: ____ (euros)
- If you think there is something important to be done (before commissioning the kitchen), do so.
- Commission the kitchen (to confirm with a kitchen seller)

3.3 Study Design

Our study used a Within-Subjects Design with one independent variable, namely the type of interaction. This independent variable has two levels: a GDI interaction and a classic DM interaction.

This Within-Subjects Design or Repeated Measures Design was necessary since we wanted each participant experienced both styles of interaction. It was important that both treatment groups exactly included the same participants. We wanted them to compare both interfaces expressing their preferences, and not to worry about personal characteristics that could bias our results.

On the other hand, in order to mitigate the potential transfer-of-learning effect caused by testing one type of interface prior to the other, a counterbalanced strategy was applied alternating the order among users. Half of the participants started with GDI, while the other half started with DM.

3.4 Measurements

Quantitative Measurements: Time on Task and Number of Errors. The first empirical data to be considered in this study is ***time on task***. In this sense, we will denote as T_{t1}, T_{t2}, and T_{t3}, the time, in seconds, that participants spent on each of the three groups of tasks, respectively. These tasks were labelled in the task sheet (section 3.2) as “Task 1: Specifying how your kitchen is,” “Task 2: Choosing furniture” and “Task 3: Retouching and ordering,” respectively. TT will be the total time, i.e. the sum of T_{t1}, T_{t2}, and T_{t3}.

The analysis of ***types of errors*** is an important and necessary complement to the timing analysis. If severe errors occurred (without the assistance of the moderator), then the time on task would be highly affected, or even worse, could prevent the user from completing the task.

Three degrees of errors have been considered, namely *slight*, *moderate*, and *severe*:

- The *severe or* blocking error took place when the user was stuck with something in the interface, preventing them from finishing the task. The moderator always assisted users with severe mistakes in order to have them finishing all the tasks and collect the corresponding user data.
- *Moderate* errors corresponded to non-blocking mistakes − not necessarily detected by the user − that could alter the outcome. In some cases, these issues could require warnings from the moderator.
- Finally, *slight* errors could not be categorised as actual errors because the user could resolve them without any moderator assistance, yet perhaps spending a bit more than the typically expected time.

Table 2 displays the concrete list of errors detected during the tests including, among other things, the type of interface and the task in which they occurred.

Subjective Measurements. We present two sets of subjective data in this paper, collected from two types of questionnaires. The first set is composed of two identical post-test questionnaires that participants filled immediately after finishing with each interface. These post-test questionnaires ask the participant seven questions (Q_i). All Q_i questions, except for Q_5, are scored with a numerical value (Table 7). The SUS (System Usability Scale) standard questionnaire was our first choice to use as the post-test questionnaire, but finally we decided to substitute its very generic questions for others more concrete and relevant for our study.

Table 2. Overview of the *Errors* (**S**:*Slight*/**M**:*Moderate*/**S**:*Severe*) specifying the interface (DM/GDI) and the tasks involved.

Type	**Interface**	**#Task**	**Incidence description**	**Effect**	**Assistance**
Slight (S)	DM	1	UNDO was confused with CLEAR when wanting to remove an item like a door (the user thought the system had failed because the item was not deleted)		
S	DM	1	The user accepted the default shape of the room; later he was aware of it and changed it		
S	DM	1	Window dimensions were confused with its distance to the walls	Put the proper window by chance	
S	DM	2 & 3	Tried to change the position of a door/window, while another operation (the placement of another door/window) was being in progress		
Moderate (M)	GDI	1	Not consciously placing a window, believing it did not matter	Consciously leaves an incomplete task (though not decisive for a correct final result)	
M	GDI	2	Instead of moving a piece of furniture (because this option was not found), deleted it and put another one at the targeted location	Inadequate procedure	
M	GDI	3	The user could not see the kitchen in 3D	Consciously leaves an incomplete task (though not decisive for a correct final result)	

M	DM	1	Entered the measurements in the default kitchen shape (without choose the correct shape), and did not realize below	Wrong final result, unconsciously	Warning
M	DM	1	The participant was unable to change the shape of the default kitchen, so he decided to start again	Inadequate procedure	
M	DM	2 & 3	UNDO was confused with CLEAR (wanting to remove a cabinet), throughout the experiment	Persistent misinterpretation of an interaction (Inadequate procedure)	
M	DM	2 & 3	Learnt not to DRAG to move furniture --> Choose delete and put it back at the targeted location	Inadequate procedure	
M	DM	3	Chosen wrong furniture (high corner instead of low)	Wrong final result, unconsciously	Warning
M	DM	3	Oblivious to the only sub-task is not explicitly stated: "choose the common elements (granite, wood and handles)"	Wrong final result, unconsciously	Warning
Looking (L)	GDI	3	With everything done, the user would choose "design a plan" (which implies starting again) instead of the "order kitchen" option, so intend to start again	It would enlarge the experiment time without limit	Help
L	DM	2	Does not learn to SCROLL (necessary to choose the desired furniture)	Blocking	Help
L	DM	2	Does not learn to DRAG (needed for "placing" the furniture)	Blocking	Help
L	DM	2 & 3	Does not learn to DELETE furniture	Blocking	Help
L	DM	3	Does not learn to DRAG (necessary to relocate furniture)	Blocking	Help
L	DM	3	Does not learn to DELETE furniture --> Try to start again	It would enlarge the experiment time without limit	Help

These questions are:

- Q_1: "Do you consider that the application has helped you in knowing *what to do* in each moment?"
- Q_2: "Do you consider that the application has helped you in knowing *how* to do it? (what you needed to do)"
- Q_3: "What was *harder?* (1) knowing *what* to do, (2) knowing *how* to get it done, (0) both easy, or (3) both hard". The scores were chosen according to a range between the best and the worst-case scenarios.
- Q_4: "Would you like to have had a more complete help system inside the application interface?"
- Q_5: "Choose (multiple choice) types of periodicity of use you consider the application is appropriate for - Just once, - Very rarely (once a year), - Once a month, - Daily"
- Q_6: "Would you use a similar application for the design of your next kitchen?"
- Q_7: "To summarize, grade how easy the application is to use"

The second set of data comes from one unique and final comparative questionnaire that participants filled at the very end, once they had used both interfaces. This questionnaire makes them directly compare both interaction styles through the next eight questions (C_i):

- C_1: "With which interface is it easier to know what to do in each moment?"
- C_2: "With which interface is it easier to know how to do it?"
- C_3: "Which interface should include more help systems?"
- C_4: "Which interface is *easier to use* and requires *less training*?"
- C_5: "Which interface lets you work *quicker*?"

- C_6: “Which interface would you recommend to a *computer professional* for an *occasional use*?”
- C_7: “Which interface would you recommend to a *kitchen design professional* for a *daily use*?”
- C_8: “Which interface would you choose for furnishing your kitchen?”

3.5 Study Procedure

The estimated time per user (including questionnaires) ranged from 45 to 75 minutes. The tests were performed individually in an interference-free environment, except for the presence of the moderator. A common laptop computer was used for all the tests. This facilitated their relocation and was less intimidating and more familiar to users than any other bigger equipment. Mouse and keyboard were the input devices. The whole process of interaction was recorded from both the computer screen and the device microphone for further analysis when necessary.

Prior to the main study, a pilot experiment was carried out. It involved a group of 4 participants. This experiment allowed us to fine-tune the test procedure, the time span, and the descriptions of each task and questionnaire.

The procedure for the study followed the next steps:

1. All participants signed an informed consent and were briefed about the fact that the tests were taken voluntarily, being free to leave at any time without any justification.
2. The users read the *moderator script* to ensure that each participant received the same information about the type of test, the purpose of the study, and the testing process. This document also clarified that these tests were not intended to make any personal or psychological assessment, but were exclusively conducted for the evaluation of the involved computer interfaces. Then, the moderator answered any participant questions or concerns, if any.
3. The participants filled out a preliminary *background questionnaire* in order to collect user characteristics, such as age, gender, computer skills, previous software knowledge, task domain knowledge, etc. This helped to check whether the participants met the screening criteria (described in section 3.1)
4. The users received a single-sheet document briefly describing the three groups of tasks to perform (detailed in section 3.2).
5. As mentioned above, in order to implement a *counterbalanced* strategy, each participant was asked to use both versions of the application (alternating the order after each user). The so-called *Thinking Aloud* technique was also applied, but in a relaxed way. The participants were not forced to explain what they were doing any time, but they were free to express themselves. The moderator tried either not to interfere at all, or to intervene only in some non-blocking situations and always in the few critical or blocking situations that took place. Specific, the moderator couldn’t answer any questions, and the users were allowed to express what they were thinking. The moderator tried not to interfere except with: a) “severe errors”, b) some "moderate errors", and c) to redirect a test. The concrete errors related to a) and b) and the type of assistance provided for each of them are shown in Table 2. In the last case, c), the moderator intervened diverting a test if the user was lasting too long on it, or if they were overwhelmed not knowing how to do something. In some of these cases, we insisted we were evaluating the interface, not themselves.

   Every error was systematically noted down and categorised as part of the data collection. For every user and for each type of interface, the time the participant spent on achieving each task was written down as well. The whole process of interaction was recorded with a computer screen and voice recording software for further re-examination when necessary. In this sense, all the measurements were checked (and corrected, in necessary case) according to a thoroughly analysis of the screen recordings.

6. At the end of the use of each interface, but before moving on to the other, the participants filled out the corresponding *post-test questionnaire* (detailed in section 3.4.2) on subjective and specific usability points about the interaction they have just experienced. This questionnaire covered important aspects for the final evaluation.
7. Finally, the participants filled out the *comparative questionnaire* (detailed in section 3.4.2) for comparison of both styles of interaction, requiring them to directly express their preferences.

3.6 Results

On the one hand, we have repeated measures designs with two conditions in our experiments because the same participant uses both types of interfaces. On the other hand, the data are not normally distributed, and then do not meet the requirements for parametric tests. Thus, to verify the significance of the results, the most appropriate statistical method is a non-parametric Wilcoxon paired-sample test, also known as Wilcoxon signed-rank test [49].

We next present the results in the following order: first, in section 3.6.1, the quantitative performance measurements through the time on tasks and the number of errors; second, in section 3.6.2, the subjective data about each interface; and, third, in section 3.6.3, the personal preferences. Finally, in section 3.6.4, we elaborate on the Eye Tracker experiment.

A comprehensive document including all the tables can be downloaded from http://bit.ly/2my1rYg

Quantitative Empirical Measurements: Time on Task and Number of Errors. Table 3 and Table 4 display the main statistics regarding *time on task* (T_{t1}, T_{t2}, T_{t3}, and T_T) and *number of errors* (`slight`, `moderate`, and `severe`), respectively.

Table 3. Statistics regarding *Time on task.*

Time on Task (seconds)								
	TASK 1: "Designing the kitchen" T_{t1}		TASK 2: "Furnishing the kitchen" T_{t2}		TASK 3: "Retouching and ordering" T_{t3}		TOTAL T_T	
	DM	**GDI**	**DM**	**GDI**	**DM**	**GDI**	**DM**	**GDI**
Min	122	120	103	136	120	105	510	429
1st Qu.	181	150	175	206	283	147	754	525
Median	**305**	**193**	**238**	**251**	**393**	**203**	**880**	**678**
3rd Qu.	410	335	524	311	516	253	1374	836
Max	1023	512	1108	560	1075	491	2591	1431
Mean	**360**	242	**363**	**276**	**436**	**214**	**1160**	**732**
σ	236	124	276	110	219	99	605	288

The corresponding Wilcoxon tests (Table 5) for T_{t1}, T_{t3} and especially for T_T, revealed significant differences ($p<0.01$) in favor of GDI, namely lower time intervals using GDI than using DM. Only the test for T_{t2}, a task consisting of repetitive operations, showed no significant difference between the two ways of interaction. Therefore, we may partially accept H1.

Table 4. Statistics regarding *number of Errors*.

	Number of Errors							
	Slight		Moderate		Severe		Total	
	DM	GDI	DM	GDI	DM	GDI	DM	GDI
Min	0	0	0	0	0	0	0	0
1st Qu.	0	0	1	0	0	0	1	0
Median	**0**	**0**	**2**	**0**	**0**	**0**	**3**	**0**
3rd Qu.	1	0	2	0	1	0	3	0.2
Max	2	1	3	1	3	1	7	1
Mean	**0.55**	**0.05**	**1.55**	**0.15**	**0.6**	**0.05**	**2.7**	**0.25**
σ	0.69	0.22	0.94	0.37	0.99	0.22	1.78	0.44

Concerning errors, the results of the *Wilcoxon* tests (Table 5) for the number of slight, moderate and severe errors produce significant differences ($p<0.02$) in favor of GDI: DM presents a significant increase in the number of errors when compared with GDI. Therefore, we can accept H2.

Table 5. Results of the Wilcoxon tests on *Time on tasks* and *number of Errors*.

Time on task	p-value	W	type of test	*#Errors*	p-value	W	type of test
T_{t1}	0.00250	180.5	one-tailed	E_{slight}	0.00750	50.5	one-tailed
T_{t2}	0.29430	134.0	two-tailed	$E_{moderate}$	0.00019	136.0	one-tailed
T_{t3}	0.00007	207.5	one-tailed	E_{severe}	0.01800	33.0	one-tailed
$\mathbf{T_T}$	**0.00001**	204.0	one-tailed	$\mathbf{E_{Total}}$	**0.00009**	171.0	one-tailed

Table 6 highlights how these evident results are pointing out by some relevant percentages: first, in accordance with T_T, 95% of users finished earlier using GDI than using the DM interface; second, 91% of the errors occurred under DM interaction versus only 9% under GDI.

Table 6. Some relevant results related to *Time on tasks* and *number of Errors*.

Interaction style	% earlier finishing	% Errors		
		Slight	Moderate	Severe
DM	5	92	91	92
GDI	95	8	9	8

Subjective Results from Post-tests Questionnaires. In section 3.4.2 we remarked that the subjective post-test questionnaire used in this experiment included seven questions (Qi) for the participants. They answered them as soon as they finished the tasks with each type of interface.

Table 7 shows some statistics summarizing the scores of each Q_i, except for Q_5 (not answered with a numerical value). According to the corresponding Wilcoxon paired-sample tests, the result for each Q_i, except for Q_5, exhibits significant differences ($p<0.001$) in the scores in favor of GDI. Therefore, we can accept H3.

Regarding Q_5, 80% of the participants considered that the DM interface is less appropriate for one-time or occasional use. Unexpectedly, 100% of the participants, including the two professionals at the task domain, considered GDI appropriate not only for occasional use but also for frequent use. Consequently, as far as these opinions are concerned, they did not find the use of guiding obtrusive.

Table 7. Statistics regarding the *post-test questionnaire* answers Q_i.

Q_i	Description	Scale	Mean		σ		Median		1st Qu.		3st Qu.	
			DM	GDI	DM	GDI	DM	GDI	DM	GDI	DM	GDI
Q_1	"Do you consider that the application has helped you in knowing what to do in each step?"	[1-7]	**3.25**	**6.30**	1.37	0.80	**3**	**6**	2	6	4.25	7
Q_2	"Do you consider that the application has helped you in knowing how to do it? (what you needed to do)"	[1-7]	**3.15**	**6.15**	1.26	0.58	**2**	**6**	3	6	4	6.25
Q_3	"What was *harder?* [1] knowing *what* to do, [2] knowing *how* to get it done, [0] both easy, or [3] both hard" The scores were chosen according to a range between the best and the worst-case scenarios".	[0-3]	**2.00**	**0**	0.51	0	**2**	**0**	2	0	2	0
Q_4	"Would you like to have had a more complete *help* system inside the application interface?"	[1-7]	**3.25**	**1.45**	1.20	0.94	**2**	**1**	3.5	1.45	4	2
Q_5	"Choose (multiple choice) types of *periodicity* of use you consider the application is appropriate for [A] Just once, - [B] Very rarely (once a year), - [C] Once a month, - [D] Daily"	Multiple choice	C+D	A+B +C+D								
Q_6	"Would you *use* a similar application for the design of your next kitchen?"	[1-5]	**3.20**	**4.65**	1.96	0.58	**2**	**4**	3	5	4	5
Q_7	"Do you consider that the application has helped you in knowing what to do in each step?"	[1-7]	**3.90**	**6.20**	1.25	0.61	**3**	**6**	4	6	5	7

Results from the Comparative Questionnaire: the User Preference. Regarding the final questionnaire composed of eight Ci comparative questions, Table 8 shows the percentage of participants preferring one way of interaction over the other. 100% of participants were clearly in favor of GDI. This might not be surprising among unskilled users, but it is remarkable in the case of kitchen designers, who are used to more completed and sophisticated DM interfaces. We next quote some samples of the opinions expressed by the participants:

- "... [GDI interface] liked me a lot for its reliance explaining it all… the other system [DM] leaves you sort of lost",
- "It doesn't matter you don't know about it, the system [GDI] tells you all along what's next",
- "I like it more the other [GDI], it's much easier; this one [DM] doesn't tell you what you have to do, you have to know it, the other guides you"

Table 8. Results of the *Comparative Questionnaire* answers. Percentage (%) of participants preferring one interaction style to the other one.

C_i	Description	DM	GDI
C_1	"With which interface is it easier to know *what* to do in each moment?"	0	100
C_2	"With which interface is it easier to know *how* to do it? (what you needed to do) in each moment?"	0	100
C_3	"Which interface should include *more help* systems?"	100	0
C_4	"Which interface is *easier to use* and requires *less training*?"	0	100
C_5	"Which interface lets you work *quicker*?"	5	95
C_6	"Which interface would you recommend for a *computer professional occasional use*?"	10	90
C_7	"Which interface would you recommend for a *kitchen design professional daily use*?"	25	75
C_8	"Which interface would you *choose* for furnishing your kitchen?"	0	100

Results from a Short Eye Tracker Analysis. Aside from the main study, we conducted a quick and simple eye tracking experiment only for two participants randomly chosen. We did not try to draw important conclusions here but simply to know the most frequent screen regions the users were looking at with each type of interface. The algorithm k-means was executed using the statistical package R, searching for hot regions manifested through point clustering. The optimum value of k for the k-means was obtained using the elbow method. This algorithm helps to automatically find, through iterative evaluation, the optimum number of clusters for a cloud of points (see, for example, [16]). This analysis showed many more points and a larger number of clusters for the DM than for the GDI type of interface. The larger number of points logically corresponds with longer time periods spent whereas the double number of clusters (12 versus 6) is associated with higher complexity. This simply reinforced our study, but as no relevant conclusions could be drawn, it was not extended to more users. The experiment results have not been included in the final numerical analysis.

4 Discussion and Conclusions

The recent shift in technology has brought about many new kinds of users who use systems occasionally in more scenarios. These are mostly users who do not want or cannot afford a learning curve. There have been many attempts to help them while they are interacting with the system. Agents, floating help windows, tooltips, direct video demonstrations, etc., have shown us the necessity of guiding the user. These help systems are ways to support the DM appealing style, but inevitably add an extra layer of supporting material over the basic direct manipulation.

We propose GDI as the basis for a completely different approach: do not expect the user to understand and use a metaphor but guide them throughout the whole process. The help system is part of the interaction; it is integrated in the interface. The user will be closely guided on the "what to do" and the "how to do," one step at a time, always according to their current goal. Consequently, these users do not have much freedom of action. They do not need to be familiar with graphical widgets or think what the next command should be. No previous knowledge is required, shortening significantly the learning curve, if any.

To support the previous assertions the authors have developed a real interface for each type of interaction. Then they have made occasional users choose which one they consider the best.

It was necessary to decide what kind of application could be appropriate for the study. The premises were: first, being an application suitable for an occasional use. Second, being an application for a widespread domain, valid for all ages, genders, and domain knowledge. And third, being an application that normally uses direct manipulation with different kinds of tasks. As a previous step, we verified that there are no interfaces other than classic DM interfaces available for kitchen design. This is a task mostly performed on an occasional basis. The design of a kitchen is a fairly open and well-known task that requires a lot of interactive objects for the user to manipulate.

When comparing both types of interfaces, three hypotheses, H1, H2, and H3, were established. The second hypothesis refers to the number of errors. It seemed natural to expect that guiding the users would prevent them from making mistakes, needing fewer moderator assistances. The obtained results have totally proven this hypothesis: for occasional use, users who are not computer experts make fewer errors using GDI than using DM.

Our first hypothesis refers to timing: non-expert occasional users proceed quicker with GDI than with standard DM. H1 may seem counterintuitive: following a sequence of steps seems to be slower than directly manipulate elements. However, the analysis of the collected data shows that GDI is more time-efficient than DM. The only task for which there was no significant difference was the repetitive Task 2.

The last hypothesis H3 has been first indirectly demonstrated from the answers of users in the *post-test questionnaires*, and then directly demonstrated from the final *comparative questionnaires*. These answers highlighted the fact that all the participants had a strong preference for GDI. For example, according to C_4, 100% of them consider that GDI is easier to use and requires less training than DM. The answers to the questions Q_1 to Q_4 and C_1 to C_3 showed the users felt guided on the "what to do" and the "how to do," which is one of the main goals of a GDI interface.

Regarding Q_5, being a question about the prospect of using both types of interfaces in the future, we should not take it but as an indication of satisfaction and never should it be taken as an accurate prediction. Still, it was unexpected to find that all participants, professionals included, considered GDI appropriate not only for occasional but for frequent use. Moreover, 80% of the participants considered that the DM interface is appropriate neither for one-time nor occasional use.

Domain professional users provided additional unexpected results: first, they managed to do the job sooner and with fewer errors with GDI; second, they showed a preference for that style. This preference could arise from the lower quality of our DM application compared to the commercial ones. But this must have been the case for both of our interfaces, not only for the DM one. In any case, the professional users did say that our DM interface had a close resemblance to commercial ones. This feedback from professionals referred only to the way they had worked with GDI. They did not consider necessary to allude to the quality of our DM version.

One of the limitations of GDI is its unsuitability for creative users or applications with many concurrent functionalities. GDI is not appropriate for users who interact creatively with applications, in such a way that the own interaction notably inspires and influences the final result. For example, a graphics editor could not be suitable for its use with a guided interface. Other examples could be spreadsheet, word processor or powerful video editing applications.

On the other hand, the specificity of the goals in less creative tasks is of utmost importance when it comes to guiding the interaction. Booking a room in a hotel, choosing a seat for our theatre performance, or applying specific transformations to a batch of files, are some examples where GDI is definitely appropriate.

We do not try here to diminish the huge importance DM interfaces have been having for decades and will have in the future. What we show here is only that an integrated guiding system as GDI is generally a better option for users who do not want or cannot afford a learning curve, offering an alternative for designers and developers of interactive systems that have occasional users as their main target users.

References

1. Americans with Disabilities. Act of 1990 incorporating the changes made by the ADA Amendments Act of 2008. http://www.ada.gov/pubs/adastatute08.pdf, last accessed 2015/08/01.
2. Annett, J., & Duncan, K.D. Task Analysis and Training Design. *Occupational Psychology*, 41, 211–221 (1967)
3. Barnum, C.M. *Usability Testing Essentials: Ready, Set...Test!* Morgan Kaufmann Publishers, Burlington, MA, USA (2011)
4. Bollaert, J. Crafting a wizard. In IBM developer Works. http://www.ibm.com/developerworks/library/us-wizard, last accessed 30/05/2015
5. Buxton, B. Multi-touch systems that I have known and loved. *Microsoft Research*, 56, 1–11 (2007)
6. Card, S., Moran, T., & Newell, A. Computer text-editing: An information processing analysis for a routine cognitive skill. *Cognitive Psychology*, 12, 32–74 (1980)
7. Card, S., Moran, T., & Newell, A. *The Psychology of Human-Computer Interaction.* Hillsdale, New Jersey: Erlbaum (1983)
8. Carroll, J.M., & Thomas, J.C. Metaphor and the cognitive representation of computing systems. *Systems, Man and Cybernetics, IEEE Transactions on*, 12, 107–116 (1982)
9. Carrillo, A.L., Falgueras J., & Guevara, A. A notation for Goal Driven Interfaces Specification. In Navarro-Prieto R., Vidal J. (eds) HCI related papers of Interacción 2004. Springer, Dordrecht (2006)
10. Carrillo, A.L., & Falgueras, J. Goal Driven Interaction vs. Direct Manipulation, an empirical comparison. In Proceedings of the XVI International Conference on Human Computer Interaction (Interacción'15). ACM, New York, NY, USA, Article 24, 4 pages (2015).
11. Charland, A., & Leroux, B. Mobile application development: web vs. native. *Communications of the ACM*, 54(5), 49–53 (2011)
12. Clarkson, J., & Coleman, R. (Eds). Special Issue: Inclusive Design. *Applied Ergonomics*, 46, part B, January, 233–324 (2015)
13. Coe, M. *Human Factors for Technical Communicators.* New York: John Wiley & Sons (1996)
14. Dix, A., Finlay, J., Abowd, G., & Beale, R. *Human-Computer Interaction.* (3th ed.). (Chapter 3). Pearson Education. Prentice Hall (2004)
15. Gong, J., & Tarasewich, P. Guidelines for handheld mobile device interface design. In *Proceedings of DSI 2004 Annual Meeting*, pp. 3751–3756 (2004)
16. Gorakala, S.K., & Usuelli, M. *Building a Recommendation System with R.* Packt Publishing Ldt. Birmingham, UK, pp. 15-18 (2015)
17. Hollnagel, E., & Woods, D. *Joint Cognitive Systems: Foundations of Cognitive Systems Engineering.* Boca Raton, FL: CRC Press (2005)
18. Hutchins, E.L., Hollan, J.D., & Norman, D.A. Direct manipulation interfaces. In D*.A. Norman & S.W. Draper (Eds.), User-centered system design* (pp. 87-124). Hillsdale, NJ: Erlbaum (1986)
19. IKEA. http://kitchenplanner.ikea.com/US/UI/Pages/VPUI.htm, last accessed 2015/02/06
20. John, B.E., & Kieras, D.E. *The GOMS family of analysis techniques: Tools for design and evaluation.* Carnegie Mellon University School of Computer Science Technical Report No. CMU-CS-94-181. Also appears as the Human-Computer Interaction Institute Technical Report No. CMU-HCII-94-106 (1994)
21. John, B.E., & Kieras, D.E. The GOMS family of user interface analysis techniques: Comparison and contrast*. ACM Transactions on Computer-Human Interaction*, 3(4), December 1996, 320–351 (1996)
22. John B.E., & Kieras, D.E. Using GOMS for User Interface Design and Analysis: Wich Technique? *ACM Transactions on Computer-Human Interaction*, 3(4), December 1996, 287–319 (1996b)
23. Johnson, P. Task Knowledge Structures. In *Diaper D. (ed.) Task analysis in Human Computer Interaction.* Ellis Horwood (1989)
24. Kieras, D. A Guide to GOMS Model Usability. Evaluation using NGOMSL. In *M. Helander & T. Landauer (Eds.), The handbook of human-computer interaction.* (2th ed.). Amsterdam: North-Holland, pp 733–766 (1997)

25. Kieras, D. Task Analysis and the Design of Functionality. In *A. Tucher (Ed.) The computer Science and Engineering Handbook*. Boca Raton, CRC Inc, pp. 1401-1423 (1997b)
26. Loranger, H. *Checklist for Planning Usability Studies.* NN/g Nielsen Norman Group. www.nngroup.com/articles/usability-test-checklist (2016)
27. Microsoft. Wizards. https://msdn.microsoft.com/en-us/library/windows/desktop/dn742503(v=vs.85).aspx, last accessed 2015/05/28
28. Moran, T.P. An Applied Psychology of the User. *Computing*, 13 (1981)
29. Mueller, J.P. *Accessibility for Everybody: Understanding the Section 508 Accessibility Requirements*. Apress (2003)
30. Myers, B., Hudson, S.E., & Pausch, R. Past, present, and future of user interface software tools. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 7(1), 3–28 (2000)
31. Newell, A., & Simon, H. *Human Problem Solving*. Prentice-Hall (1972)
32. Nielsen, J. *Usability engineering*. San Diego, California, United States of America. Academic Press (1993)
33. Nielsen, J., & Budiu, R. *Mobile Usability*. New Riders (2012)
34. Nielsen. J. *Quantitative Studies: How Many Users to Test?* NN/g Nielsen Norman Group. www.nngroup.com/articles/quantitative-studies-how-many-users/ last accessed 2015/11/17
35. Norman, D.A. *The Design of Everyday Things.* Basic Books, Inc., New York, NY, USA (2002)
36. Norman, D.A. Natural user interfaces are not natural interactions. *Interactions*, 17(3), 6–10 (2010)
37. Paternò, F. ConcurTaskTrees: An Engineered Notation for Task Models. In: *D. Diaper, & N. Stanton, (eds.), The Handbook of Task Analysis for Human-Computer Interaction*, ch. 24, pp. 483–503. Lawrence Erlbaum Associates, Mahwah (2003)
38. Payne, S.J., & Green, T.R.G. Task Action Grammar: recent developments. In *D. Diaper (Ed.) Approaches to Task Analysis*. Cambridge University Press (1990)
39. Savidis, A., & Stephanidis, C. Unified user interface design: designing universally accessible interactions. *Interacting with Computers*, 16(2), 243–270 (2004)
40. Selker, T. Touching the future. *Communications of the ACM*, 51(12), December (2008)
41. Sharples, M. The design of personal mobile technologies for lifelong learning. *Computers & Education*, 34(3), 177–193 (2000)
42. Shneiderman, B. Direct Manipulation: A Step Beyond Programming Languages. *IEEE Computer*, 16(8), 57–69 (1983)
43. Shneiderman, B., Plaisant C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. *Designing the User Interface: Strategies for Effective Human-Computer Interaction.* (6th ed.). Pearson (2016)
44. Sova, D.H., & Nielsen, *J. How to Recruit Participants for Usability Studies.* NN/g Nielsen Norman Group. www.nngroup.com/reports/how-to-recruit-participants-usability-studies, last accessed 2017/10/28
45. Tidwell, J. *Designing Interfaces: Patterns for Effective Interaction Design.* O'Reilly Media Inc. Gravenstein Highway North, Sebastopol, CA 95472 (2011)
46. Turoff, M. The design and evaluation of interactive systems. Section 1.4.3: "User Roles and Types" http://web.njit.edu/~turoff/coursenotes/IS732/book/tablecon.htm, last accessed 2014/05/23.
47. US Rehabilitation Act Amendments Section 508, 1998. http://www.section508.gov/ last accessed 2015/10/01
48. Wickham, D.P., Pupons, D., Mayhew, D.L., Stoll, T., Toley III, K.J., & Rouiller, S. *Designing Effective Wizards: A Multidisciplinary Approach*. Upper Saddle River, NJ: Prentice Hall (2002)
49. Wilcoxon, F. Individual Comparisons by Ranking Methods. *Biometrics,* 1*,* 80–83 *(*1945)