THE USABILITY OF SOFTWARE SYSTEMS OR HOW HARD IS TO MAKE THINGS SIMPLE

Adriana-Mihaela Tarța*

**Babeş-Bolyai University, Faculty of Mathematics and Computer Science,*
Str. Mihail Kogălniceanu nr. 1, 400084 Cluj-Napoca, Romania
Tel: +40-264-405327, fax: +40-264-591906
adriana@cs.ubbcluj.ro

Abstract: The use of computer systems has encountered a widespread in the last years in most of the domains of our lives. Every year, the number of people using the computer in their everyday tasks is increasing and, proportionally, the number of people experiencing difficulties in using and understanding the computer increases. Moreover, many businesses discover that the systems in which they have invested a lot of resources remains unused because they don't really support the users in their tasks. The umbrella under which all these problems can be discussed is called usability, and the present paper presents some methods to evaluate the usability of software systems and proposes an improvement of a model used to predict users' behavior in front of an interface.

Key words: usability, patterns, cognitive, heuristics, usability metrics

1. INTRODUCTION

We are used to see in every place we go computers supporting people in their everyday tasks, but sometimes we also see frustrated users when it comes to use the systems that should help them in their tasks. It's a reality that still most of the software systems remain idle or impose long training courses in order to make the users perform their tasks using the systems conceived to help them in their work. A new discipline has appeared in the early eighties, called Human-Computer Interaction (HCI) with the aim to study the interaction between human and computers and to give solutions to the problems encountered. The usability of computer systems is a very actual and complex problem and a new branch of HCI has emerged during nineties, called Usability Engineering.

Section 2 of the this paper defines the notion of "usability" and presents some possible measures of usability. Section 3 briefly presents the methods used to evaluate the usability of a system, emphasizing on GOMS model and suggesting the use of another models (laws) to improve the results of the analysis. At the end of this section it is presented an alternative to estimative models of usability represented by the interaction patterns. These patterns can be used when designing the user interface and their appliance grow the designer's confidence regarding the usability of the system being designed.

2. WHAT IS USABILITY?

The usability problem has become subject for standardization, so that ISO 9241-11 gives us the following definition of usability: “The usability is the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use” [1]. These standards relate to usability as a high level quality objective. The effectiveness is the accuracy and completeness with which specified users can achieve specified goals in particular environments. The efficiency refers to the resources expended in relation to the accuracy and completeness of the goals achieved. The satisfaction refers to the comfort and acceptability of the work system to its users and other people affected by its use.

The above definition does not explicitly specify operational criteria that might lead to an understanding of what we should evaluate when it comes to usability.

The quality attributes of software are measured using metrics. Table 1 presents an example of usability metrics that can be used in order to evaluate the usability of a system:

Usability Objective	Effectiveness Measures	Efficiency Measures	Satisfaction Measures
Suitability for the Task	Percentage of goals achieved	Time to complete a task	Rating scale for satisfaction
Appropriate for trained users	Number of "power features" used	Relative efficiency compared with an expert user	Rating scale for satisfaction with "power features"
Learnability	Percentage of functions learned	Time to learn criterion	Rating scale for "ease of learning"
Error Tolerance	Percentage of errors corrected successfully	Time spent on correcting errors	Rating scale for error handling

Table 1- An example of usability metrics

In other words, usability addresses the relationship between tools and their users. The tools are effective when they allow their users to accomplish their tasks in the best way possible.

3. HOW CAN BE USABILITY ACHIEVED?

The key principle to maximize the usability of software systems is the use of the iterative design which progressively refines the design through evaluation from the early stages of design and taking into account users’ feedback when making the evaluation.

There are a variety of methods for the evaluation of the usability of a software system. In choosing a specific method one should keep in mind the following aspects: the cost of the evaluation, the appropriateness of the method for the project, the time constraints of the project, and the cost associated with the implementation and training of new users.

Cognitive walkthrough is a method for evaluating an interface based on breaking down and analyzing actions that a user must perform in order to use the system or perform a task. Cognitive walkthroughs are performed at any stage of the design using a prototype, a conceptual design document or the final product.

Focus groups gather groups of users to get their feedback, initial reactions to a design, and discuss their preferences. Focus groups are good at discovering the points where the system tested differs from the user's expectations.

Prototyping involves developing representations of a system for testing and evaluation. Prototyping is an essential element of an iterative design approach, where designs are created, evaluated, and refined with the results of testing at each cycle fed into the design focus of the next cycle.

Usability inspection is a review of a system based on a set of guidelines. The review is conducted by a group of experts who are deeply familiar with the concepts of usability in design.

User testing observes actual users interacting with the system. Users are asked to perform tasks while usability experts observe and take note of their actions. User testing is the best method when it comes to finding usability problems.

Heuristic evaluation is a technique for finding usability problems, the evaluation being based on a set of heuristics [6]. The evaluation is made by few trained evaluators who inspect the user interface individually, than the results are compared and a set of conclusions is formulated.

Task analysis evaluates how the end-user actually uses the software systems.. An analyst determines the user's goals and tasks, then makes recommendations aimed at increasing efficiency and user-friendliness. In the following paragraphs a task analysis method, called GOMS, will be presented.

From all the usability metrics presented in Table 1, the time is the easiest to measure. The next method is preoccupied on estimating the time to complete a task and the way of predicting the time will be presented shortly in the following lines.

GOMS (Goals, Operators, Methods and Selection Rules) [3] is a user modeling technique aimed to provide reasonably accurate predictions about the human behavior in front of an interface.

A *goal* is a symbolic structure that defines a state of affairs to be achieved and determines a set of possible methods by which it may be accomplished.

Operators are elementary perceptual, motor, or cognitive acts whose execution is necessary to change any aspect of the user's mental state or to affect the task environment. Operators are assumed not to be concurrent, so only linear processes can be modeled.

A *method* describes a procedure for accomplishing a goal. It is one of the ways in which a user stores his knowledge of a task

When a goal is attempted, there may be more than one method available to the user to accomplish the goal. *Selection rules* represent the control structure in the model that describe the possibility of choosing one of more alternatives to achieve a goal.

GOMS analysis works by breaking the task in a stack of goals and by specifying the operators, methods and selection rules used to choose between alternate ways of achieving the goals. By doing this it is possible to predict the users' routes when performing a task. Additionally, *the GOMS keystroke level model* [4] helps the designer in making predictions about the time required for users to perform the series of gestures needed in the interaction with a software system. In the interaction with the computer the user uses a set of fundamental gestures, each such gesture needs a time to be performed. The notation introduced for representing the time needed to perform the fundamental gestures is: K –

keying, representing the time to perform a keystroke or a mouse click; P – *pointing*, representing the time to position the mouse pointer; H – *homing*, the time required for user to move his hands from the keyboard to the mouse; M – *mental*, representing the time for the user to prepare for the next step; R – *responding*, representing the time for the computer to respond to the user inputs. The fundamental principle is that the total time to perform a sequence of gestures is the sum on the individual gestures [3]. The authors of the model have used typical values for time need to perform the fundamental gestures experimentally determined (K = 0.2 sec., P = 1.1 sec., H = 0.4 sec., M = 1.3 sec.). Because the users are so different, these values may vary a lot (even 100%), so that the time estimation may be affected. Using this way the model it is possible only to determine trends and compare user interfaces, but not estimating with accuracy the time needed by user in order to achieve a task. In order to solve this problem, in this paper I propose the use of some laws that estimate the time needed to perform some of the fundamental gestures previously enumerated (K, P, M). The use of these laws should improve the use of GOMS analysis regarding the estimation of the time needed to perform a task, because they take into account more factors affecting the interaction. In the next sections I will present the laws that I found interesting to apply for a more accurate estimation of fundamental gestures previously presented.

3.1 Fitts' Law

Fitts' Law is a model to account for the time it takes to point at something, based on the size and distance of the target object.

$T = k * \log_2\left(\frac{D}{S} + 0.5\right)$, k ~ 100 ms, where:

T = time to move the hand to a target ;
D = distance between hand and target ;
S = size of target [8].

Fitts' Law and its variations are used to model the time it takes to use a mouse or other input devices to click on objects on the screen. Broadly, Fitts' Law can be applied by designers to suggest moving target buttons closer and making them larger for extremely commonly used buttons. Fitt's Law can be used for a more accurate estimation of P when using the GOMS keystroke level model in order to predict the user's behavior in the interaction with a system.

3.2 Meyer's Law

Meyer's Law is a refinement of Fitts' Law for predicting the time it takes for rapid aimed movements, such as hitting a button on the screen by moving a mouse to it (K, P).

$T = A + B * \sqrt{\frac{D}{W}}$, where:

T = time to move to a target;
D = distance to target;
W = width of target;
A ~ -13 ms;

B ~ 108 ms [5].

A and B are constants which may vary with the input device. Meyer's Law can be used to make predictions of how much time it will take for a user to accomplish a task involving selection of targets on the screen (such as icons, menus, or hypertext links).

3.3 Hick's Law

Hick's Law estimates the time needed to make a decision (M).

(1) $H = \log_2(n + 1)$.

(2) $H = \sum_{i=1}^{n} p_i * \log_2\left(\frac{1}{p_i} + 1\right)$, where:

H = the information-theoretic entropy of a decision;

n = the number of equally probable alternatives;

p_i = the probability of alternative *i* for *n* alternatives of unequal probability [8].

The time it takes to make a decision (M) is roughly proportional to H, the entropy of the decision (the log of the number of alternatives), i.e. T = kH, where k ~ 150 ms. This can be used to make a time estimate for how long people will take to make a decision in using a user interface, such as choosing a menu item or selecting an item on a navigation. The time M depends on the knowledge of user on the consequences of choosing an alternative and on the way that the alternatives are presented to the user.

3.4 The Power Law of Practice

The Power Law of Practice is an alternate way of estimating the time needed by a user to perform a task taking into account the expertise level of the user in accomplishing a task using the evaluated system. The power law of practice is an expression of the time required to perform a task, based on practice trials. The base idea is that the users improve their performance after some practice trials.

$T_n = T_1 * n^{-a}$, a ~ 0.4 , where:

T_n = the time to perform a task after n trials;

T_1 = the time to perform a task in the first trial;

n = the number of trials [8].

GOMS uses this law in order to make predictions of human performance.

The use of the methods mentioned above may be expensive and time consuming, but still we wish to make our systems usable. An alternative may be the use of interaction patterns in the design of the user interfaces. The idea of using patterns in the design of user interfaces is taken from software engineering and previously from architecture.

3.5 Interaction patterns

A pattern is a solution to a problem that appears over and over again. Interaction patterns for user interfaces are the result of the effort of interface developers to overcome the problems that users have when interacting with systems. The patterns take in consideration the usability principles of Nielsen and each pattern presents the problem to which it addresses, the usability principle on which the solution is based, the context of use, the solution, the rationale to use the pattern, an example and, eventually, an counterexample [7]. Applying interaction patterns doesn't exclude the need to evaluate the user interfaces, but only assures that some usability problems will be avoided, although improving some aspects of usability may imply an negative effect on other aspects of usability.

4. CONCLUSIONS

Usability is an essential quality of user interfaces and must get a lot of attention from the designers of the systems. From the users' point of view, the usability is important because it can make the difference between performing a task accurately and completely or not, and enjoying the process or being frustrated. From the developers' point of view, the usability is important because it determines the success or failure of the system. In this paper some approaches of usability and some solutions to improve the usability of systems have been presented. Also, the paper have presented a method of merging the results of the research in the area of Human-Computer Interaction in order to improve the results obtained when using a GOMS analysis as a method of evaluating the design of a user interface.

The technical progress has led to a situation where not only desktop computers can run user interfaces, but also palm and smart phones are capable of running user interfaces, and the interaction devices may vary from a platform to another (a mouse may be present, or not). It's important now to find new ways, more general, to describe and to evaluate the interaction between the user and the system, independent of the platform and the device used, because the users are interested in using the system they know in the same way, no matter the device they use to interact with the respective system.

The aspects presented in this paper reveals the fact that assuring usability of systems may be a very complex task, but the effort is worth because keeping the users satisfied is one of the most important goals when talking about software systems, and business generally.

5. REFERENCES:

1. Bevan N., (1995), *Human-Computer Interaction Standards,* Proceedings of the 6th International Conference on Human Computer Interaction, Yokohama, Anzai & Ogawa (eds), Elsevier;
2. Booth P., (1989), *An Introduction to Human-Computer Interaction*, Lawrence Erlbaum Associates, Publishers;
3. Card S. K., Moran T. P., Newell A., (1980), *The Keystroke-level Model for User Performance with Interactive Systems*, Communications of the ACM, 23:396-410;
4. Kieras D., (1994), *GOMS Modeling of User Interfaces using NGOMSL. Tutorial Notes*, CHI'94 Conference on Human Factors in Computer Systems, Boston, MA, April 24-28;
5. Meyer D.E. et al., (1988), *Optimality in human motor performance: ideal control of rapid aimed movements,* Psychological Review, 95:3, 340-370;
6. Nielsen J., (1992), *Finding usability problems through heuristic evaluation*, Proceedings ACM CHI'92 Conference (Monterey, CA, May 3-7): 373-380;
7. Troetteberg H., Welie M., (2000), *Interaction Patterns in User Interfaces*, PloP 2000;
8. *Usability glossary*, http://www.usabilityfirst.com