Aspects of User Behaviour Modelling

József Tick

John von Neumann Faculty, Budapest Polytechnic, tick@bmf.hu

Abstract: A well designed and attractive user interface is one of the key points of the success of software systems. It is necessary and practical to modell user's behaviour in the process of a system development that meets all the requirements to the highest possible. However, it is not enough to modell user's behaviour at the level of implementation and take it into account only at the level of design. Tracing the user's behaviour should be an integral part of a system and a change in these behavioural habits must be monitored automatically and continuously by the system. This solution means that a system must have an adaptive user interface that continously analyses the user's behaviour and adapts to the user's demand.

Keywords: Software Engineering, User Interface Design, Human Computer Interaction

1 Introduction

In the old days the primary goal of software development was the system's functionality. From the user's point of view, the correct and failure free operation of a system was of the highest importance. Nowadays, however, in line with the above mentioned focus points, the quality of communication between the software and its user has become of the same importance. Easy handling, easy learnability and its integration into already existing systems have also been extreemely highlighted. The appearance and spreading of these requirements made the systematic design of user interface more and more important. Beyond the design of user interface, a change in these requirements had a great impact on functionality, primarily on requirements connected to the operational speed and response time. An exact specification of requirements is necessary in case of software design that is up to the requirements. In case of user interface, however, the requirements are often not unified, since on the one hand a system is used by several types of customers, while on the other hand these requirements can be measured only with difficulty. A solution can be to modell the user's behaviour and to carry out a design fitting into this modell.

2 Rate of design in user interface development

There are four phases of design in the user interface development:

1. Lack of design. Those developments are grouped here in which the user group and the user modell are not determined at all.
2. User = developer viewpoint. The developments in this group are characterised by the developer regarding him/herself both the average user and the user modell at the same time.
3. Determination of the user group. In this category the user group is determined during the development, however, the modell setting is at a rough guess and taken from the simple average, while the testing, the success examination and the evaluation of user interface are not carried out.
4. The thorough design of user interface. In this category, after setting up a well based modell before the development, continuous testing, success evaluation and user satisfaction analyses are carried out during and at the end of development.

As this paper urges, the aim is to design a user interface modelled in point 4., since the marketability of software products and the high level satisfaction of customers can be secured only in this way.

3 Methods of user behaviour modell definition

Modelling of user's behaviour includes two fundamentally different steps. The first one is described as the definition of the initial modell, since this modell is set up in the course of the implemetation phase, before the design of the user interface, and it provides a base for the user interface design. The second step is the definition of the adaptivity of the user interface and its integration into the system. This system possessing this adaptive feature continuously collects information about the user's behaviour during its usage, it evaluates them and finally on the basis of the results it adapts to the user's changed behavioural habits by changing the user interface.

3.1 Problems at defining user behaviour modell

Software systems appear in front of a user through the user interface. It means that for the people the user interface reflects the software itself, so this is the thing that can be experienced from the whole system during its usage. It means that the same

software system will be differrent for each user, furthermore, thanks to motivational, emotional and cognitive processes (learning, routine, adaptivity, success and failure experience) developing in the user during the usage of a system, the same user will gradually regard the same software different while making progress in its usage.

The below two features [1] of the Human-Machine-Environment system modell, namely the physical features (body size, etc.) and the physiological features (muscle strength, etc.) can be ignored from the aspect of software systems. The values of the further three elements of the system modell, namely the perceptual features (sight, colour sight, space sight, etc.), the cognitive features (memory, thinking, etc.) and the motivational and emotional features are determining factors in the aspect of modelling the behaviour of the users who handle different software systems.

However, the measurement, evaluation and analyses of the three above listed features can be put into numbers with difficulty and is a quite complicated task. The spreading of the user group makes the situation more difficult. The most difficult case occurs when the spreading is quite large, that is when users with very different age, qualificaton, intelligence and cultural background handle the system. A good example can be a public system designed for a wide spread audience. Especially the group of users of web-based applications is difficult to be characterised, since customers from all over the world can use the services of a system.

3.2 A possible method to define a modell

A modell can be defined by getting to know the user's behaviour. Considering the factors described in the previous chapter it can be concluded that the user will be a hypothetical "average" user, whose behaviour will be considered as the average member of the entire user group, furthermore, to whom the modell will be adjusted.

An "average" user can modell a deformed user behaviour as well and since it is not a real user, "only a modell", it can help at the evaluation during the design but it cannot be used at the testing of the system under development. It means that parallel to the "average" user, "blood and flesh" users are also needed, i.e. a few-membered but representative user group is needed, which group can help in the testing, controlling and evaluation of a system.

Considering the above mentioned aspects, the realisation of a modell is carried out as written below:

1. Determining the whole user group

2. Selecting a representative pattern on the basis of the examination of the whole user group
3. Determination of the parameters of a hypothetical user using the evaluation results of the testing of the representative group, defining the initial user behaviour modell

3.3 Determination of modell elements

Before starting the development of user interface it is fundamental to prepare a user modell. This modell includes all those features and information (gender, age, schools, qualification, motivation, etc.), which can be known about the user. This modell contains information gained from different questionnaires and tests.

The above described modell does not include, and cannot include the dynamic behaviour of a user. These features can be learnt only via computerised environment, while the usage of special test systems and test environments are necessary to gain this information.

The definition of a dynamic modell is done by getting to know the user's behaviour. Considering the factors described in the previous chapter, it can be concluded that a user will be a hypothetical "average" user, whose behaviour will be considered as the average member of the entire user group, furthermore, to whom the modell will be adjusted.

The measurable and valid behavioural features that can be used while modelling are listed below:

3.3.1 The average reaction time in decision situation

The user's reaction time in decision situation (choosing between menue points) is a feature with several components, which includes the orientation security on a given site, the perception speed, the decision time and the actual execution time.

Although the evaluation and the analyses of these features can be carried out independently, in a lot of cases it is enough to evaluate the average speed of system handling in order to characterise the user.

This feature can be measured and evaluated by the handling speed of a well specified test environment, that is by the measurement of the execution time of individual tasks, and by the examination of the spreading and the average of the gained time figures.

3.3.2 The degree of usage learning

During the usage of the system, the user learns to handle the system. This can be realised in three figures:

1. a decrease in the average execution time of test environmental tasks
2. a decrease in the number of errors during usage
3. a decrease in the frequency of the usage of help tools

With the measurement and comparision of these features during the repetition of tests it can be examined how fast the user is capable to learn the handling of a system, and after which test cycle the number of handling errors drops below an acceptable level.

3.3.3 The rate of oblivion

The user who has already gained some practice but does not use a certain system for a while will naturally forget, and "lose learnt skills". Thus the features mentioned above get worse. The rate of decline per unit time can be measured during the tests.

Oblivion can be measured in such a way that a time factor is monitored below which the decline of the parameters of handling reaches a disturbing level.

3.3.4 The rate of navigational safety

One of the important components of the fast and effective handling of the system is the navigational safety. Of course, navigatinal safety increases with the rising frequency of the system usage, furthermore, the logical mapping of the menue structure that also reflects the given subject field will reduce the initial problems to a great extent making the system user friendly and easily managable.

In addition, however, it is necessary that users have preliminary knowledge of navigation. This preliminary knowledge means partly professional, partly computer knowledge and partly general features. The rate of preliminary knowledge and its stability provide important information for the developer about the system user.

3.4 Integrate adaptivity into the system

The modell described above reflects the user's behaviour during the usage of a system. In the light of a change in his/her behaviour, an adjustment in certain features of a system to the changed behaviour would increase both the effectivity of handling and the feeling satisfaction with the system.

In case of values characterised by time, an integral part of the user interface can be such a subsystem that continuously measures the values of the parameters of the features listed in 3.3, while in case of a significant distortion, time and speed factors are modified.

An other useful adaptive feature can be an effective navigational help for the user.

The user takes a determinable route from the start menue point to the target menue point. By overlapping these routes like graphs a center of gravity route can be determined, that occurs with a certain frequency in all of the possible registered routes. If the occurrence of these routes is significant we can prepare our software system with the application of different techniques to help visitor to go through the center of gravity route in the most simple way possible.

This means that on the basis of both the already walked routes and the user's behavior pattern, we try to find out the user's target menue point and offer the possibility to walk out the with the system "hand in hand" on the optimal route to reach the searched funtionality.

Conclusions

The shaping and forming of the user interface of software systems have a great impact on the opinion and satisfaction about a certain software product. The application of the user behaviour modell can make the software poduct successful in the long run, since the system will be capable to react to behavioural changes due to the user's skills and learning.

The method described in this paper is a possible way to consider a user behavioural modell during software development.

References

[1] Izsó, L., Antalovits, M.: Bevezetés az információ-erdonómiába, BME, 2000.

[2] Shneiderman, B.:Designing the User Interface, Addison-Wesley, 1992.

[3] Tick, J.: User Interface Redesign based on User Behaviour Analysis, ICCC 2003.