

Influences of Software Postures on Software Interface Design

Chao Gao^{1,a}

¹Room 7632, Building7, Tian Garden, Kunming University of Science and Technology, Kunming, Yunnan Province, China

^aplayerchao@126.com

Kunjian Wang^{2,b}

²Faculty of Arts and Communication, Kunming University of Science and Technology, Kunming, Yunnan Province, China

^b1601292016@qq.com

Abstract—By exploring aesthetics in human-computer interface applications, thus summed up, how to improve computer interactive interface to enhance the beauty of the user when the efficient use of human-computer interaction and the use of feelings.

Keywords-Software Posture; Software Design; Interactive Design

I. INTRODUCTION

In GB, the software is defined as computer programs, procedures and rules related to computer system operations, as well as possible files, documents and data. Software posture means the behavioral posture and position shown to users. All software has their primary ways to expose to users. For example, the software may be bold or timid, black, white or colorful. Through postures we can discuss how much attention users may pay to the interaction of the software and how the software behavior correspondingly responds to this attention. Notably, paper software posture is not determined by the preferences of designers and programmers, and the software interface design must also base on the understanding of real situations and environments to make truly meaningful design.

A number of important principles in the design section are led by the software posture, but the software posture is not a black-or-white and discernible thing. Software may exhibit different postures and a variety of characteristics, which is related to factors such as the usage scenarios of users. In this paper, the author will outline three software postures and discuss users' real use behaviors about each posture and the principles of design.

II. SOFTWARE POSTURE

In general, interaction design is originated in desktop software. In recent years, the interaction design concept has been extended to the other devices. This paper will focus on desktop software posture features. The general posture of desktop software can be divided into three: exclusive posture, transient posture and background posture. Every gesture describes a different behavior collection, as well as different types of user behaviors. The summary of the desktop software postures can give designers an important basis for interface design.

A. Exclusive Posture

If the software occupies users' attention for a long time, it is called exclusive posture software. Exclusive posture software can provide a series of related functions and features, and users often need to keep these features showing up and running, taking up the entire screen. The most typical example of exclusive posture software is the text-editing software, forms software and many vertical applications, which often occupy the entire screen for a long time. Interaction of these programs is generally very complex, and users are often in a state of workflow when using exclusive posture programs. In addition, exclusive posture software often runs in a maximized window. For example we use the word rarely in a reduced window because in that case we will not be able to use many common features. In short, the biggest feature of exclusive posture software is that it can be used by users and used continually for a long time. They have become a primary tool for users and dominated the users' workflow. For example, when users use Powerpoint to make slides, they will use it in full-screen mode from the start to the end. Maybe sometimes they need other programs to help finish the job, but Powerpoint always maintains its exclusive posture.

When using exclusive software, people tend to spend a lot of time and energy. Such a long time usage can lead us to the conclusion: users of exclusive software are often middle users, not novices or experts in this software. Therefore, from this perspective, designers should do more to improve the design for this group of users. It means that the design of such software doesn't need to focus on beginners, which will make the software clumsy and inflexible; at the same time, they should not pursue sophisticated toolssimply. Anyway, the user type targeted by the product design must be the primary ones among software users. Among the middle users of the exclusive software, many people just use it occasionally, which shouldalso not be ignored by the designers.

Because the interaction between users and exclusive software occupies users' time, the software needs to make full use of screen space. During this period of interaction time, there is no other software to compete with the exclusive software. Hence, designers can design software toolbar as needed without the need for deletion. In case of

other posture, a combination of multiple toolbars may make the software too complex, while the exclusive software has a great advantage in the use of screen pixels. In the interface design process of this kind of software, the following design methods can be used:

(1) Minimal Visual Style

As users will always stare at the exclusive software when using it, so designers should weaken the visual expression of color and texture, etc. At the same time, please do not use too many colors, which will make the software tacky. The designers can use a stronger shade when emphasizing or highlighting some content.

(2) Rich Visual Feedback

Exclusive application is a good platform for it can create an excellent visual feedback environment for users. The designers can add some meaningful visual indications and suggestive contents for user behaviors, such as status bars, data mode and program state. In addition to ensuring the rich visual feedback, designers should avoid interface confusion. Reasonable use of visual feedback can make users more skilled and satisfied with the software.

(3) Rich Input

Exclusive application can also enjoy a store of input methods. The common functions in the software can also be operated in many ways, such as direct operation, dialogue box, keyboard shortcuts.

Exclusive software should fill the screen, which is also applicable for the document window in the program, i.e. the child window of the document should always be maximized in the process of program use unless users have different requirements.

Much exclusive software is centered on documents, but this not means that all the document-related software is exclusive application. If a program performs a single function when operating documents, it is not exclusive application software and it should not show some exclusive behaviors as this kind of software shows the transient posture.

B. Transient Posture

The transient posture software will be immediately shut down after a short while of operation, and it only shows a single function to users. When the exclusive software is working, some transient posture software can be invoked and help users when necessary but leave after finishing works, thus making users continue to work with exclusive software.

The greatest characteristic of transient posture software lies in its temporality. The software will not stay on the screen for a long time, so users will not be familiar with them. As a result, this kind of software requires display controls carefully and clearly. This kind of software does not need strong artistry and it requires an accurate, big icon and easily read big font to show the content clearly.

Although transient posture applications can run independently on the desktop, it is usually used to support the exclusive application. The following is a typically example of transient scene: when using the Word (an office software), you need to open the Explorer to open some

other contents. Hence, when designing transient posture software, you should consider the existence of exclusive software and make sure the transient posture software does not occupy too much screen space.

Although the transient posture application needs to save screen space, the control button on the interface should be larger and more eye-catching. Unlike the exclusive application which leads to users' visual fatigue, the transient posture application just stay on the screen temporarily. Therefore, the large and bold button helps users locate and use it conveniently. The use frequency of the transient posture application is low, thus it is easy for users forgetting the meaning of each function key. Therefore, the instructions of the transient software should be displayed on the surface and the button functions should be expressed as clearly as possible.

Generally, only a window and a view are applied in the transient posture software. In the design of transient posture software, the scroll bar and some subtle mouse actions should be avoided as they are not suitable for this kind of software.

Any transient posture software may be invoked when using exclusive posture software. As the former is on top of the latter, this means the transient posture software will block some information of the exclusive software. Therefore, the transient software should be movable and there should be an easy place for its dragging and moving.

No matter for transient posture software or exclusive posture software, the best way to help users is to equip program with memory ability. For example, the transient software can remember use state of the last time, or the previous position and configuration, thus help users in the next time. This way is better than any of the default settings.

C. Background Posture

Background posture software refers to the software having no interaction with users. This kind of software working behind can complete tasks, even crucial tasks, without user intervention. Network connection and printer driver are such kind of software. As long as the computer is open, this kind of software will perform the process and it must be ready to be loaded or adjusted casually. The interaction between users and background software is temporary, so the design principle of transient posture software is also applicable for background posture software.

The background posture is very important for applications. Normally, the application is invisible, so the way of calling programs is also very important sometimes. In the windows system, the most common way is to show them by application icons which are usually displayed in the lower right. In order to save table space, versions after windows xp hide the occasionally used background programs behind. Both the windows and mac systems configure the background software by control panel, so users can stop or start the background software.

III. SUMMARY

In sum, the design of software should be user-oriented. The interface of the software will affect users' manners; the relationship between interfaces and users also affect the usability of the software; the interface and the use behavior should not against the final destination. It can be seen that the interface is important to both users and designers.

The designers should take the software posture into consideration. Based on this orientation, designers carry out late design, research and development. Only under this premise, the software can provide a better user experience.

REFERENCES

- [1] Alan Cooper, Robert Reiman, David Cronin. About Face3. Beijing: Electronic Industry Press,2012.: 2-255
- [2] Steve Krug. Don't Make Me Think. America: New Riders Press, 2005: 55
- [3] Donald Norman. Design Psychology. Beijing: China CITIC Press, 2007: 42-98
- [4] Jakob Nielsen. How Users Read on the Web. America: Academic Press. 2004: 10-38
- [5] Christina Wodtke, Austin Govella. Information Architecture: Blueprints for the Web. Beijing: Posts and Telecom Press, 2009-11: 40
- [6] RobertHoekman. Designing the Obvious: A Common Sense Approach to Web and Mobile Application Design, Second Edition. Beijing: China Machine Press, 2008-1.37-64
- [7] Peter Merholz / Brandon Schauer / David Verba / Todd Wilkens.The Power of Design. Beijing: China Machine Press, 2009: 3-13
- [8] Lukas Mathis. Designed for Use: Create Usable Interfaces for Applications and the Web. Beijing: Posts & Telecom press, 2012: 32-36