Marton Sakal

GUI vs. WUI Through the Prism of Characteristics and Postures

Article Info:

Management Information Systems,
Vol. 5 (2010), No. 1,
pp. 003-006

Received 12 Jun 2008
Accepted 24 April 2009

UDC 004.492.514

Summary

Because of the platform differentiation of software products today, the dominant styles of interfaces are: GUI (Graphical User Interface), WUI (Web User Interface) and HUI (Handheld User Interface). Similar goals, principles, rules, heuristics are valid for every of the listed styles at the conceptual level, while substantial differences are present at the level of the physical design. The domination trend favors GUI and its derivatives. GUI dominates on the PC platform and, in essence, there are a small number of its variations. WUI dominates in the field of the approaches to the Internet, extranet and intranet. The HUI design mostly depends on the extent to which extent the PDA device supports the GUI style and the size of the display. The work presents the basic characteristics of GUI and WUI interface styles, their similarities and important differences.

Key words

GUI, WUI, interface design, interface patterns

1. Graphical user interface

The graphical user interface can be defined as the style of interaction in the user-computer route, based on four fundamental elements – the initial letters of these elements give the acronym WIMP, a frequently used synonym for GUI: Windows, Icon, Menus, Pointers. The cited elements make a semantic and functional entirety in the association with the direct manipulation. The **direct manipulation** is probably the most important characteristic of GUI. It enables the user the interaction with objects by the use of the pointer device (the typical examples include drag-and-drop, text selection with the mouse, creating objects in graphical programs, and so on). Many operations available by the menu system are also realizable by means of direct manipulation. For example, if we "put down" the icon representing the Word document on the icon representing the MS Word program, it is the equivalent of the action where the Word is read, then the menu item File⇨Open is activated and the document is selected.

Thanks to the direct manipulation, **the user experiences the system as the extension of the real world -** supposing that the user is familiar with the objects and actions over these objects in the real world, replicating objects and actions (methods) by means of GUI artifacts, the user is enabled to work in the familiar environment in the familiar way, where the focus from the application and its tools, making the physical organization of the computer system invisible, moves to the data themselves, i.e. tasks which the user wants to perform. **The constant visibility of objects and actions is present** – like the objects on the desk, virtual objects and actions performed are visible on the display continually and in real time.

The concept of direct manipulation is not strictly associated with GUI. Some early text editors already had some characteristics of direct manipulation, but the appearance of graphical environment has enabled the implementation of this principle in the true sense of the word. Nevertheless, not every object is subject to the direct manipulation for different reasons, such as complexity of conceptualization of such an operation in the graphical environment, limited possibilities of the graphical system, limited area of the window, and so on. In these cases, indirect manipulation is applied. Most of today's GUI systems are the combination of direct and indirect manipulation. For example, the menu item is available by the click on the desired menu option (direct manipulation), and the menu itself is the textual operation list (indirect manipulation).

Discussions about the advantages and disadvantages of some user interface styles often turn into anecdotes, losing gradually the scientific level in favor of emotionality. Therefore, excluding details, some of advantages and disadvantages will be described below.

GUI advantages. The graphical information presentation has many possibilities of utilization of the user's capabilities for information processing. Regularly implemented, GUI reduces the need for perceptual and mental recoding and reorganizing thus reducing the intensity of using memories, and increasing the speed of information transfer in the man-computer interactive relation. Some further advantages of GUI are: decrease in user error rate when using the GUI software product, expressive sense of control, predictability of the system responses and behavior, reversibility of performed actions, decreased anxiety of beginners, increased attraction and better esthetical experiences, decreased volume of difficulties in translation from one language into another because of the increased use of universal graphical symbols, drastically reduced need to use keyboards and so on.

GUI disadvantages. Besides many positive advantages of GUI, the graphical presentation of interactions is not always an optimum choice, but the classic one, on the text based user interface represents a more effective and efficient solution. In these situations, forced conversion of all systematic components in the graphical format can result negatively. Also, graphical interfaces become more and more sophisticated, more complex, and sometimes even bizarre. Systemizing objections, we can list the following disadvantages of GUI:

- The number of GUI components and techniques of their application available to the GUI designer are constantly increasing. Because of the increased number of alternatives and design artifacts, software products with experimental graphical interface are increasingly offered on the market, and the learning time of initial assumptions of using graphical environment has increased.
- There is some inconsistency in the terminology and techniques of GUI development, as the consequence of evasion of possible legal implications associated with copyright and the attempts to differentiate software products by means of unique user interfaces (similar to the practice present at the time when DOS-based programs dominated).
- Limitations of human perceptive and cognitive capabilities are expressed fully with users' interfaces with many icons. The number of different symbols that one person can differentiate and effectively manage when performing tasks by means of a software product is more limited than in the textual presentation. The fact that the recognition of pictograms requires a lot of perceptual learning and capability of abstraction and intelligence is often neglected.
- In contrast to typography with 300 year of experimenting, icon design is drastically limited by extremely short period of time available for marketing of a new software product.

1.1. Typical postures of the graphical user interface

Based on the postures, i.e. the way of representing the contents to users and carrying out interactions, the graphical user interface can be of: sovereign type, transient type, daemonic type, auxiliary type (Cooper & Reimann, 2003).

Sovereign postures are mostly and best used in full screen mode and their type is MDI (Multiple Document Interface), where child windows are used maximized. Such interfaces offer more competitive functions and monopolize the user's attention for a long period of time. Typical examples are text processors, spreadsheet programs, e-mail applications, and so on. These programs are typically primary tools for performing users' jobs and their productive use requires intermediary knowledge and experience.

Transient postures usually serve as support, supplements to sovereign postures, and are typically monothematic and present a restrictive set of controls (mostly one function). These postures are invited according to the needs and they are used very shortly, after that the user returns to sovereign postures. Because of the emphasized temporality, users rarely become familiar with them, and therefore interface controls are, as a rule, stressed, bigger and homogeneous; they do not require highly developed users' motoric skills. They rarely use maximized windows and have sub windows very exceptionally. Usually, there is no need to provide the possibility to change the size of windows, but all the more so, the change of position should be enabled and so provide the visibility of part of the interface (window) contents of the sovereign type. The keyboard interface leads down to the buttons Enter, Tab, Esc and navigation keys. The typical example is the Calculator.

Daemonic postures are program interfaces which very rarely perform interaction with users (that is the reason why they are experienced by typical users as esoteric) because they move automatically together with starting the system, they are performed imperceptibly, in the rear, carrying out vital systemic tasks with no use to intervene. Typical examples are drivers for printers and programs whose visual represents can be found in the system tray.

Auxiliary postures are a combination of sovereign and transient postures: they are permanently visible, but only play the role of support. Task-bar is a good example. The programs with such interface are mostly quiet process reporters that are performed at the moment. Their messages should be simple and respect maximally the interface of sovereign postures.

2. Web user interface

If GUI has basically changed the way of creating user interface, it can be said that the Web has revolutionized computation providing software users with previously unimaginable freedom and control in using it, for example, by the possibility to change typographies, colors, leaving them decisions whether to present or not the graphics on the read Web page, express agreement to forward or not data by insecure channels of communications, accept or refuse cookies, and so on. In a strikingly short time period, an average user has become intuitive for good design, not accepting the slow web page reading, confused navigation, illogical page organization, uneasy animations, and so on. The possibility of cheap, simple and fast change of web locations has created, from an average user, a Web voter who makes decisions about the destiny of websites by clicking the mouse.

However, as long as an insignificant difference is present in the philosophy and the path of interfaces of browsers themselves, at the same time, the real anarchy reigns in the interface design of web applications performed in the browser windows. It is known that the Web was planned as a communication medium of scientists and researchers, but it has grown commercial arms in companies and homes all over the market-guided global world. However, it is less known that the Web is a victim of the omission in establishing the basis of the HTML language and unexpected and explosive (maybe haphazard) development. The underlying HTML language is not conceived as the language by means of which screens and contents, available to the broad and technically inadequately educated population, will be created. The complexity of designing web pages is not only the repercussion of shortages of the HTML language but also the fact that the logic of linear navigation in browsers originates from the pre-era of informatics development. There are a lot of problems with back-to-where and forward-to-where functions, and also the use of Back icon at an inconvenient time can destroy a lot of work. The characteristics of one of the most useful GUI elements, windows, in the Web design cannot be used; there is not a simple or elegant solution for secondary windows, as an example.

The WUI design is the design of navigation, in fact, and information presentation where the attention is paid to the contents. The user does not move between applications (as with GUI) but from one site to the other. The primary purpose of web pages is to present useful information, but some anarchy is present here. Very often, contents which the user neither searches nor wants are presented. For navigation, the basic WUI style is mostly similar to the hierarchical menu structure in non-graphical environments.

The interface design has a significant role in creating such menu hierarchies and contents of pages which should enable the natural feeling and ease of use. Therefore, designers, unfortunately, often create agnostic environments where:

- Applications present information by the use of many browser windows;
- There are no conventional icons and components but different web applications use different icons and animations for specific navigation or purely esthetic reasons; it often leads down to the so-called visual noise, visual pollution and unacceptably long time for page reading;
- Pointers support basically the action of the simple click in selection and navigation; drag-and-drop is not systematically incorporated but it is realized by additional programming.

2.1. Web posture

Information-oriented sites are intended primarily for presenting information without any need to perform transactions. The interface of these sites usually balances the need to present useful information with the need to enable new and irregular visitors to learn and navigate easily.

The interfaces **of web applications** differ if the great majority of users visit them daily within the framework of regular, work tasks (B2B) or these visits are less frequent. In the first case, the interface mostly takes on the characteristics of sovereign postures, and in the second one, the combination of sovereign, temporary and auxiliary postures are applied because they are equally important, as well as the detailed review of information and navigation.

Web portals. In the early Web days, search engines were web portals in the true sense of the word. They provided the user with the possibility to move to another web location by the mouse click. These portals are called **navigation portals** today. For users who rarely visited these portals, the temporary posture was appropriate, with the clear, simple navigation control. The auxiliary posture was appropriate for regular visitors. However, more integrated contents were offered by developing these portals. Therefore, consumer-oriented portals enable a unified approach to contents and functionalities, associated with specific topics, while enterprise portals enable internal approach to important corporate information and business tools. In these cases, the interface should create the environment where users can have access to a particular kind of information and perform a particular kind of work, i.e. the environment which enables that information collected from different sources be presented in one place. These portals are called **environmental portals**. In this way, as these portals create the working environment, the impression of the place is created so portals are no longer the way to go to another place, but they become destinations. The appropriate posture for these portals is the sovereign one, but it can be brought down to the composition of temporary postures.

Marton Sakal
University of Novi Sad
Faculty of Economics Subotica
Segedinski put 9-11
24000 Subotica
Serbia
Email: marton@ef.uns.ac.rs

Conclusion

Only the use of GUI is not an automatic guarantee of the software product success. Yet, GUI-based programs provide better usability, more effective initial learning and generally higher productivity in solving real problems. Besides, consistently implemented GUI platform standards enable transfer and mobility of learning and knowledge in different GUI software products. Also, it is an obvious tendency that WUI becomes more and more similar to GUI, probably because of positive experiences and habits of users formed in using the GUI environment.

References

Cooper, A., & Reimann, R. (2003). *About Face 2.0.* New York: John Wiley & Sons.

Đurković, J., & Tumbas, P. (2000). *Metodološki prilazi, metodi i tehnike razvoja informacionih sistema.* Subotica: Ekonomski fakultet.

Galitz, O. (2002). *The Essential Guide to User Interface Desing.* New York: John Wiley & Sons.

Jošanov, B., & Tumbas, P. (2002). *Softverski inžinjering.* Novi Sad: Viša poslovna škola.